Sepa exactamente cómo una computadora CZ Spectrum puede prolongar su cerebro.

1. Si usted es médico puede llevar la agenda de turnos de sus pacientes. Guardar los datos de cada uno y obtener así las liquidaciones para las Obras Sociales.



2. Si tiene una Agencia de Prode, quiniela y loterías, puede registrar la venta total de la semana; cuáles son los números más pedidos, los atrasados y los ganadores del Prode.
Con una CZ SPECTRUM, tendrá la grande en su local.



3. Si tiene un autoservicio puede mantener actualizado su stock de mercaderías, la rotación del mismo y ordenar su facturación.

4. Si tiene una agencia de autos, podrá llevar un registro completo de cada unidad; color, modelo, kilometraje y hasta modificar el precio al instante.

5. Si es constructor podrá tener rápidamente un panorama completo de cada obra. La compra de materiales por rubro, cálculos estadísticos, cantidad de personal, índice de rentabilidad.
Con la CZ SPECTRUM, comienzan hoy los cimientos de sus obras del año 2000.

6. Si administra consorcios, ingrese los gastos de cada edificio y obtenga en instantes el cálculo de las expensas.

7. Si vende pasajes podrá tener un control absoluto de todos los vuelos. Teniendo en cuenta los distintos tramos. Compañías. Tarifas en dólares y la transformación automática quincenal y mensual.

8. Si administra clubes, lleve el registro de socios.

9. Si tiene un restaurante, registre fácilmente los cambios de la lista de precios, ingrese los consumos por mesa y obtenga las adiciones a pedido.

10. Si tiene una inmobiliaria, lleve el registro de las propiedades en venta. Hasta qué suma puede estudiar una oferta. Qué zona le falta cubrir para ofrecer. Dónde están las propiedades. Qué características tienen. Ubicación, acceso, comodidades, precio.

11. Si su actividad es el comercio exterior, con la CZ SPECTRUM tenga en segundo datos fundamentales para su negocio; cantidad de toneladas cosechadas, precio del grano en los distintos mercados, cantidad de cabezas ingresadas a Liniers en los últimos años, su disponibilidad de mercaderías exportables al instante. CZ SPECTRUM inclina la balanza a su favor.

12. Con la CZ SPECTRUM un aventajado estudiante como usted, resuelve rápidamente cálculos, ecuaciones o fórmulas y tiene a su alcance cientos de datos de cualquier materia. Y luego, para despejar la mente, corra una carrera de Fórmula 1, rechace un ataque extraplanetario o simplemente dé un paseo en su avión. CZ SPECTRUM la simplificación del estudio.

13. Y cualquiera sea su actividad, con la CZ SPECTRUM, la computación empieza a formar parte de su vida de todos los días para hacerla más fácil.
Porque eso es la tecnología CZ: computación al alcance de todos.

CZ Spectrum

La prolongación de su cerebro

Adquiérala en los Distribuidores de la



Distribuye en todo el país. CZERWENY ELECTRONICA S.A. Av. de Mayo 963 - 3º P. (1084) Capital Federal

Bergoli & Valentini

MICROCOMPUTADOR MICRODIGITAL TK-90X

Color y sonido
a través del T.V.
16K y 48 K

EL MICROCOMPUTADOR
CON MILES DE PROGRAMAS



GARANTIA 6 MESES

En venta en comercios de microcomputadores, artículos del hogar, electrónica, fotografía y librerías.

SOFTWARE Y PERIFERICOS TOTALMENTE COMPATIBLES CON ZX SPECTRUM +" ®

- Control del volumen del sonido a través del TV (sintetizador operado por BASIC)
- Interface incorporado para joystick
- Mensajes de ejecución y código de reportes de errores en castellano.
- TRACE: Comando de seguimiento de programas, permitiendo la rápida corrección de errores de lenguaje.
- UDG: Comando de editor de caracteres especiales definidos por el usuario (acentos, Ñ, etc.).
- Feedback sonoro del teclado
- Fuente de alimentación con interruptor.
- Ameno, fácil y completo manual de instrucciones en castellano.

MICRODIGITAL

Importa, distribuye y garantiza:
ARVOC s.a.i.c.f.i.
Avda. DIAZ VELEZ 4149 (1200) Capital Federal Tel.: 981-1980/9212

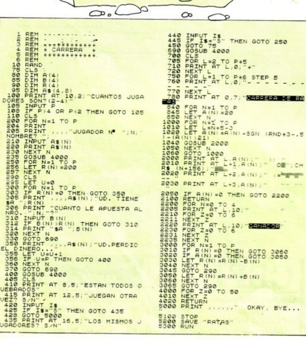
CARRERA DE RATAS

COMP: CZ1000/1500 TK83/85
CONF: 16 K
CLAS: ENT

Este juego sólo ocupa 4 K de memoria RAM. Cada jugador posee una laucha y 200 pesos para apostar (usted sólo puede apostar a su rata).

Si su rata gana, entonces obtendrá la misma cantidad que arriesgó, y pasará a aumentar su capital. Si pierde, le será descontada la apuesta. Si se queda en cero, queda fuera del juego.
Una vez que todos los jugadores se hayan quedado sin dinero, el juego comienza otra vez.





```
   1 REM --------- ---
   2 REM -   .  .   .
   3 REM ************
   4 REM * CARRERA  *
   5 REM ************
   6 REM
  10 RAND
  75 CLS
  80 DIM A(4)
  85 DIM B(4)
  90 DIM R(4)
  95 DIM A$(4,8)
 100 PRINT AT 10,2;"CUANTOS JUGA
DORES SON?(2-4)"
 105 INPUT P
 108 IF P>4 OR P<2 THEN GOTO 105
 110 CLS
 200 FOR N=1 TO P
 204 PRINT
 205 PRINT ,,,,"JUGADOR Nº ";N;"
NOMBRE? ";
 210 INPUT A$(N)
 220 PRINT A$(N)
 230 NEXT N
 235 GOSUB 4000
 250 FOR N=1 TO P
 256 LET R(N)=200
 257 NEXT N
 290 CLS
 295 LET U=0
 300 FOR N=1 TO P
 303 IF R(N)=0 THEN GOTO 350
 305 PRINT ,,,,A$(N);"UD. TIENE
$A ";R(N)
 306 PRINT "CUANTO LE APUESTA AL
NRO. ";N;"?"
 310 INPUT B(N)
 312 IF B(N)>R(N) THEN GOTO 310
 314 PRINT "$A ";B(N)
 320 NEXT N
 330 GOTO 690
 350 PRINT ,,,,A$(N);"UD.PERDIO
EL DINERO..."
 355 LET U=U+1
 356 IF U=P THEN GOTO 400
 360 NEXT N
 370 GOTO 690
 400 GOSUB 4000
 405 CLS
 410 PRINT AT 8,5;"ESTAN TODOS Q
UEBRADOS"
 415 PRINT AT 12,5;"JUEGAN OTRA
VEZ? S/N"
 420 INPUT I$
 425 IF I$="S" THEN GOTO 435
 428 GOTO 5000
 435 PRINT AT 16,5;"LOS MISMOS J
UGADORES? S/N"
 440 INPUT I$
 445 IF I$="S" THEN GOTO 250
 450 GOTO 75
 690 GOSUB 4000
 700 CLS
 705 FOR L=2 TO P*5
 710 PRINT AT L,0;"*"
 720 NEXT L
 750 FOR L=1 TO P*6 STEP 5
 760 PRINT AT L,0;"- - - - - - -
- - - - -"
 770 NEXT L
 780 PRINT AT 0,7;"CARRERA DE RA
TAS"
 840 FOR N=1 TO P
 845 LET A(N)=20
 850 NEXT N
1000 FOR N=1 TO P
1010 LET L=N*5-3
1020 LET A(N)=A(N)-SGN (RND*3-.5
)-(A(N)>21)
1040 GOSUB 2000
1050 NEXT N
1060 GOTO 1000
2000 PRINT AT L,A(N);"  () "
2010 PRINT AT L+1,A(N);"  O";CH
R$ (N+156);"  "
2020 PRINT AT L+2,A(N);" <=
-- "
2030 PRINT AT L+3,A(N);"  "" ""
"
2050 IF A(N)=0 THEN GOTO 2200
2100 RETURN
2200 FOR N=0 TO 4
2205 PRINT AT L,10;"        "
2210 FOR Z=0 TO 6
2211 NEXT Z
2220 PRINT AT L,10;"GANADOR"
2230 FOR Z=0 TO 6
2231 NEXT Z
2235 NEXT N
3000 FOR N=1 TO P
3010 IF R(N)=0 THEN GOTO 3060
3020 IF A(N)=0 THEN GOTO 3050
3030 LET R(N)=R(N)-B(N)
3040 NEXT N
3045 GOTO 290
3050 LET R(N)=R(N)+B(N)
3060 NEXT N
3065 GOTO 290
4000 FOR Z=0 TO 50
4010 NEXT Z
4020 RETURN
5000 PRINT ,,,,,,"  OKAY, BYE...
"
5100 STOP
5200 SAVE "RATAS"
5300 RUN
```

CALENDARIO 2

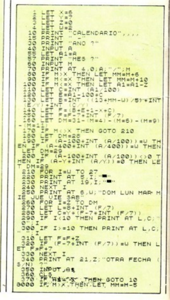
COMP: CZ1000/1500 TK83/85
CONF: 16 K
CLAS: PER

Es un programa tan útil como divertido, pues le permite visualizar la hoja calendario en el mes y año que desee, dentro del siglo XX.

PANTALLA

```
CALENDARIO
 -
ANO ?
1985/7
DOM LUN MAR MIE JUE VIE SAB
     01  02  03  04  05  06
 07  08  09  10  11  12  13
 14  15  16  17  18  19  20
 21  22  23  24  25  26  27
 28  29  30  31

 OTRA FECHA (S/N) ?
```



```
   1 LET X=6
   2 LET Y=7
   3 LET Z=0
   4 LET W=2
  10 PRINT "CALENDARIO",,,,
  15 PRINT " -"
  30 PRINT "ANO ?"
  40 INPUT A
  55 LET A1=A
  57 PRINT "MES ?"
  60 INPUT M
  70 PRINT AT 4,0;A;"/";M
  80 IF M>X THEN LET MM=M+6
  90 IF M<=X THEN LET MM=M+10
 100 IF M<=X THEN LET A1=A1-Z
 110 LET C=INT (A1/100)
 120 LET E=A1-100*C
 130 LET B=INT ((13*MM-W)/5)+INT
 (E/Y)+INT (C/Y)
 140 LET F=(B+E+1-X+C)
 150 LET F=F-7*INT (F/7)
 160 LET DM=31-(M=4)-(M=6)-(M=9)
-(M=11)
 170 IF M<>X THEN GOTO 210
 180 LET DM=28
 190 IF (A-100*INT (A/100))=W TH
EN IF (A-400*INT (A/400))=W THEN
 LET DM=29
 200 IF (A-100*INT (A/100))<>0 T
HEN IF (A-Y*INT (A/Y))=0 THEN LE
T DM=29
 210 FOR I=W TO 27
 220 PRINT AT 5,I;"■"
 230 PRINT AT 19,I;"■"
 240 NEXT I
 250 PRINT AT 6,W;"DOM LUN MAR M
IE JUE VIE SAB"
 260 FOR I=Z TO DM
 270 LET L=8+INT (F/7)
 280 LET C=Y*(F-7*INT (F/7))
 290 IF I<10 THEN PRINT AT L,C;"
0";I
 300 IF I>=10 THEN PRINT AT L,C;
" ";I
 310 LET F=F+Z
 320 IF (F-7*INT (F/7))=W THEN L
ET F=F+7
 330 NEXT I
 340 PRINT AT 21,Z;"OTRA FECHA (
S/N) ?"
 350 INPUT A$
 360 CLS
 370 IF A$="S" THEN GOTO 10
8000 IF M>X THEN LET MM=M-5
```

CAZA FANTASMAS

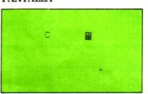
- COMP: CZ 100/1500 TK83/85
- CONF: 2K
- CLAS: ENT

INSTRUCCIONES:

Este es un programa de persecución en el que hay que evitar encuentros con los fantasmas ("), y comer la mayor cantidad de píldoras de fuerza (*); El jugador sale representado por una C y puede moverse por toda la pantalla, usando las teclas con flechas. El puntaje aumenta cada vez que coma una píldora, y aparecerá recién cuando lo atrape un fantasma.



PANTALLA



```
   1 REM    ****  K 64  ****
   2 REM    *CAZA  FANTAMAS*
   3 REM    ****************
  10 LET S=0
  20 LET A=0
  30 LET B=0
  40 LET C=20
  50 LET D=30
  60 LET E=INT (RND*10+1)*2
  70 LET F=INT (RND*15+1)*2
  80 LET A$=INKEY$
  90 LET A=A+(2 AND A<20 AND A$=
"6")-(2 AND A>0 AND A$="7")
 100 LET B=B+(2 AND B<30 AND A$=
"8")-(2 AND B<0 AND A$="5")
 110 LET C=C+(A>C)-(A<C)
 120 LET D=D+(B>D)-(B<D)
 130 CLS
 140 PRINT AT A,B;"C";AT C,D;"▀"
AT E,F;"*"
 150 IF A=E AND B=F THEN GOTO 18
0
 160 IF C=A AND B=D THEN GOTO 20
0
 170 GOTO 80
 180 LET S=S+1
 190 GOTO 60
 200 PRINT AT 0,0;"TU PUNTAJE ES
";S
 210 STOP
 220 SAVE "C.F"
 230 RUN
```

BASKETBALL

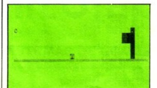
Comp: CZ 1000/1500
Conf: 2 K
Clas: Entretenimiento



INSTRUCCIONES:

Hay que embocar la pelota en el cesto. En pantalla el jugador sale representado por una "A" inversa, que se mueve de izquierda a derecha, presionando cualquier tecla lanzarás la pelota.
El cesto saldrá arbitrariamente en cualquier lado de la pantalla, y tu puntaje aparecerá en el vértice superior izquierdo de la pantalla.

PANTALLA



```
   1 REM    **** K 64 ****
   2 REM    * BASKETBALL *
   3 REM    **************
   4 LET S=PI-PI
   5 FOR V=SGN PI TO CODE "▒"
   6 CLS
   7 LET U=PI-PI
   9 LET Q=INT (RND*CODE "?")+CO
DE "$"
  10 FOR G=U TO CODE "▌"
  11 PRINT AT CODE "▖",VAL "G+5"
;"▒▒▒▒▒▒";AT G,0;"█"
  12 NEXT G
  13 PRINT AT U,U;S;AT PI/PI,Q-C
ODE "▪";"██",TAB Q-CODE "▪";"██"
  20 FOR F=U TO Q-CODE "▪"
  21 PRINT AT CODE "▌",F;" ▣"
  23 IF INKEY$<>"" THEN GOTO COD
E "M"
  30 NEXT F
  49 GOTO VAL "102"
  55 FOR D=CODE "▪" TO PI-PI STE
P -SGN PI
  57 LET F=F+SGN PI
  60 PRINT AT D,F;"O",AT D,F;" "
  63 NEXT D
  80 FOR D=PI-PI TO CODE "▌"
  81 IF U=PI-PI THEN LET F=F+PI/
PI
  82 IF D=CODE "▪" AND VAL "F+1"
=Q THEN LET U=PI/PI
  83 IF U=PI/PI THEN LET S=S+VAL
".2"
  84 IF F>=CODE "3" THEN GOTO VA
L "102"
  85 PRINT AT D,F;"O";AT D,F;" "
  86 NEXT D
 104 NEXT V
 105 PRINT AT PI-PI,PI-PI;S
 108 STOP
 109 SAVE "BASK"
 110 RUN
```

SUELDOS TEMPORARIOS

Comp: CZ1000/1500 TK83/85
Conf: 16 K
Clas: Comercial



Con este programa se podrá calcular sueldos en forma temporaria.
El programa brindará:

- Total por horas trabajadas.
- Total de descuentos (jubilación, cuota sindical, etc.)
- Sueldos a cobrar.

Al comenzar el programa pedirá:

- Pesos por hora.
- Horas trabajadas en total.
- Horas trabajadas en días feriados.
- Horas de ausencia por enfermedad.
- Cómo fue el "presentismo".

```
   1 REM  ******  K 64  *******
   2 REM  **********************
   3 REM  *SUELDOS TEMPORARIOS+
   4 REM  **********************
  10 CLS
  11 LET T$=""
  12 LET N$=""
  13 LET Y$="--------------------------------"
  14 LET E$="ENTRE DATOS"
  15 LET X=0
  16 LET Y=21
  17 LET X$="                                "
  18 LET L=16
  20 GOSUB 500
  22 PRINT Y$
  25 GOSUB 600
  30 GOSUB 700
  35 GOTO 872
  40 CLS
  42 GOTO 1008
 500 CLS
 501 PRINT AT 1,0;"NOMBRE ";N$;AT 1,17;"TARIFA $";T$
 502 RETURN
 600 PRINT AT 9,1;"HRS.TRABAJADAS "
 610 PRINT AT 11,1;"HRS.FERIADO"
 620 PRINT AT 13,1;"HRS.ENFERMEDAD"
 630 PRINT AT 15,1;"PRESENTISMO"
 640 PRINT AT 17,1;"ADELANTO"
 650 RETURN
 700 REM RUTINA DATOS DE ENTRADA
 705 PRINT AT 1,7;"?"
 707 PRINT AT Y,X;E$
 710 INPUT N$
 715 GOSUB 1000
 720 PRINT AT 1,7;N$;AT 1,26;"?"
 725 PRINT AT Y,X;E$
 730 INPUT T$
 735 GOSUB 1000
 740 PRINT AT 1,26;T$
 745 LET TA=VAL T$
 750 PRINT AT 9,L;"?";AT Y,X;E$
 755 INPUT A$
 760 GOSUB 1000
 765 PRINT AT 9,L;" ";AT 9,29-LEN A$;A$
 770 LET HT=VAL A$
 775 PRINT AT 11,L;"?";AT Y,X;E$
 780 INPUT A$
 785 GOSUB 1000
 790 PRINT AT 11,L;" ";AT 11,29-LEN A$;A$
 795 LET HF=VAL A$
 800 PRINT AT 13,L;"?";AT Y,X;E$
 805 INPUT A$
 810 GOSUB 1000
 815 PRINT AT 13,L;" ";AT 13.29-LEN A$;A$
 820 LET HE=VAL A$
 825 PRINT AT 15,L;"?";AT Y,X;"ENTRE (SI/NO)"
 830 LET A$=INKEY$
 832 IF A$="S" THEN GOTO 840
 833 IF A$="N" THEN GOTO 835
 834 GOTO 830
 835 GOSUB 1000
 836 PRINT AT 15,L;" ";AT 15,27;"NO"
 837 LET P=X
 838 GOTO 850
 840 GOSUB 1000
 845 PRINT AT 15,L;" ";AT 15,27;"SI"
 846 LET P=1
 850 PRINT AT 17,L;"?";AT Y,X;"ENTRE (SI/NO)"
 852 LET A$=INKEY$
 853 IF A$="S" THEN GOTO 860
 854 IF A$="N" THEN GOTO 856
 855 GOTO 852
 856 GOSUB 1000
 857 PRINT AT 17,L;" ";AT 17,27;"NO"
 858 LET ADE=0
 859 GOTO 870
 860 GOSUB 1000
 861 PRINT AT Y,X;"ENTRE MONTO"
 862 INPUT ADE
 863 GOSUB 1000
 864 PRINT AT 17,L;" ";AT 17,23;"$";AT 17,29-LEN STR$ INT ADE;ADE
 870 PRINT AT Y,X;"DATOS OK? (SI/NO)"
 871 RETURN
 872 LET A$=INKEY$
 874 IF A$="N" THEN GOTO 880
 876 IF A$="S" THEN GOTO 40
 878 GOTO 872
 880 CLS
 881 LET N$=""
 882 LET T$=""
 883 GOTO 20
1000 PRINT AT Y,X;X$
1002 RETURN
1008 FAST
1009 GOSUB 500
1010 PRINT Y$
1011 LET TH=0
1012 LET LL=15-LEN STR$ INT TA
1014 LET TT=INT (TA*HT*100)/100
1020 PRINT AT 3,1;"HRS.TRAB.  X ";AT 3,LL;TA;AT 3,28-LEN STR$ INT TT;TT
1022 LET TH=TT
1024 LET TF=INT (TA*HF*100)/100
1026 PRINT AT 4,1;"HRS.FER.   X ";AT 4,LL;TA;AT 4,28-LEN STR$ INT TF;TF
1028 LET TH=TH+TF
1030 LET TE=INT (HE*TA*100)/100
1032 PRINT AT 5,1;"HRS.ENF.   X ";AT 5,LL;TA;AT 5,28-LEN STR$ INT TE;TE
1034 LET TH=TH+TE
1037 IF P=1 THEN LET P=INT (TH*8.334)/100
1038 PRINT AT 6,1;"PRESENTISMO";AT 6,28-LEN STR$ INT P;P
1040 LET TH=TH+P
1050 PRINT Y$
1060 PRINT AT 8,1;"TOTAL HABERES ";AT 8,28-LEN STR$ INT TH;TH
1062 PRINT Y$
1064 LET TD=0
1065 LET D=INT (TH*11)/100
1066 PRINT AT 10,1;"JUBILACION 11";AT 10,28-LEN STR$ INT D;D
1068 LET TD=D
1069 LET D=INT (TH)/100
1070 PRINT AT 11,1;"LEY 19032 1";AT 11,28-LEN STR$ INT D;D
1071 LET TD=TD+D
1074 LET D=INT (TH*3)/100
1076 PRINT AT 12,1;"LEY 18610 3";AT 12,28-LEN STR$ INT D;D
1078 LET TD=TD+D
1080 LET D=INT (TH*2)/100
1082 PRINT AT 13,1;"CUOTA SIND. 2";AT 13,28-LEN STR$ INT D;D
1084 LET TD=TD+D
1086 PRINT Y$
1088 PRINT AT 15,1;"TOTAL DEDUC. ";AT 15,28-LEN STR$ INT TD;TD
1090 PRINT Y$
1092 PRINT AT 17,1;"IMPORTE LIQUIDO";AT 17,28-LEN STR$ INT (TH-TD);TH-TD
1094 PRINT AT 18,1;"ADELANTOS";AT 18,28-LEN STR$ INT ADE;ADE
1096 PRINT AT 19,1;"A COBRAR";AT 19,28-LEN STR$ INT (TH-TD-ADE);TH-TD-ADE
1097 SLOW
1098 PRINT AT Y,X;" COPIE Y LUEGO PRESIONE ENTER"
1099 LET A$=INKEY$
1100 IF CODE A$<>118 THEN GOTO 1099
1101 LET N$=""
1102 LET T$=""
1103 GOTO 20
8999 STOP
9000 SAVE "HABERES"
9901 CLS
9910 PRINT "SISTEMA SUELDOS TEMPORARIOS"
9920 FOR J=1 TO 90
9930 NEXT J
9940 RUN
```

C U R S O S

CURSOS BASIC I AVANZADO
ASSEMBLER - Profesores especializados
COMMODORE 64 - APPLE II

Av. PUEYRREDON 2034 (1119)
BUENOS AIRES - Tel. 84-7663



CURSOS especializados para usuarios de todas las marcas.

Cupo Máximo
12 personas por clase
COMIENZA "JUNIO"
Edad 11 años en adelante
DURACION: 3 MESES

MANIAC:
Rivadavia 13734 Ramos Mejía
(1704) Tel.: 654-6844

MAGIC LOADER
Solución definitiva para los problemas de carga de programas

CONVERSION PAL-N
TIMEX SINCLAIR 2068

ELECTROSOUND
Laboratorios Electrónicos

Viamonte 1336
Piso 8º Of. 48
Tel.: 45-8585 - Capital

CDI CENTRO PARA EL DESARROLLO INFORMATICO

CURSOS DE COMPUTACION

DESDE 1 A 3 PERSONAS POR COMPUTADORA
ENSENANZA PERSONALIZADA - PRACTICA PERMANENTE
CUOTA MENSUAL ₳ 15,- DESCUENTO JULIO 17%

C.D.I. AV. SANTA FE 1714 1º PISO - GALERIA FRANCIA - CAPITAL.

APRENDA COMPUTACION
EN UNA EMPRESA DE COMPUTACION
CON GENTE DE COMPUTACION

- CURSOS TEORICOS-PRACTICOS
- GRUPOS REDUCIDOS
- EQUIPOS DISPONIBLES PARA PRACTICAS
- POSIBILIDAD DE BECAS RENTADAS

INFORMES E INSCRIPCION:
PTE. R.S. PEÑA 950. CAPITAL TEL.: 35-6582/6465

PROMUEVEN: Q.B.S.A. Y SUPERMICRO S.A.



unicomp s.r.l.

SISTEMAS DE COMPUTACION

Distribuidores
Microdigital
Latindata
ZX Spectrum
Timex Computer 2068
Acoustech (Grabadoras para computación)
Pelikan (Cintas para impresoras)
VISICOMP (Monitores B y N, Verde)

Diskettes
Maxell
Pelikan
Datalife
FUJI
SKC

Bibliografía
Textos
Revistas
(Nacionales e Importadas)

Cassettes (Programas)
TK 85
Microsoft TK 90
TK 2000
Spectrum
Commodore
Sinclair 2068

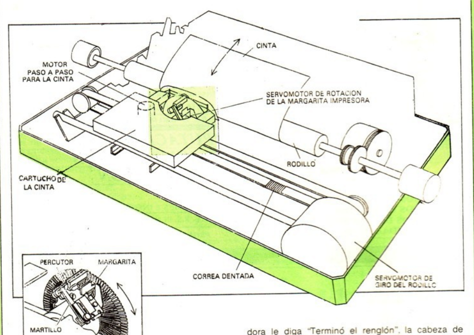
Impresoras
OKIDATA
u 82 A microline
Alphacom 32

Accesorios
- Interface de grabador para Commodore 64
- Codificadores de señal para computadoras
- Joysticks
- Formularios continuos

A partir de Agosto iniciación Cursos de Programación
una computadora por alumno

MONROE 4502 esq. LUGONES Tel. 51-2754/2659

La Impresora



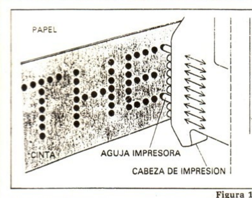
Es quizás el periférico mas usado en todo tipo de computadoras. Evidentemente por más exacto y rápido que funcine un computador, si no se pueden representar sus datos de salida en una forma clara y cómoda para su uso humano, de nada servirá. Es por este motivo fundamental que adquieren gran importancia la forma en que estas salidas se imprimen.
Para conocer una impresora vamos a explicar su funcionamiento interno, para despues tratar de hallar una clasificación funcioal de las mismas. Cuando el computador quiere comunicarse con una computadora, le puede mandar los datos en paralelo o en serie (Ver K64 Nº 3 Pag. 42). Existen impresoras que pueden recibir los datos de las dos formas y otras que precisan alguna interface especial pero, sea como sea que lo reciba, la impresora almacena el dato en su memoria interna.
Algunas impresoras tienen una memoria interna grande, donde almacenan miles de bytes para después imprimirlos; pero por lo menos la memoria interna de la impresora es de un renglón, de forma tal que cuando se llena el renglón o cuando la computadora le diga "Terminó el renglón", la cabeza de impresión se mueve y escribe una línea completa.
La forma en que la computadora le dice "Terminó el renglón" es a través de un código formado por un byte, interpretado según una tabla llamada de código ASCII.
El código ASCII, es universal y permite comunicar las computadoras de todo el mundo con una gran gama de periféricos y entre sí.
Este mismo código permite saber para cada byte del código qué letra le corresponde, de forma tal que debe haber en la memoria interna una memoria permanente (ROM), que trae grabada la información necesaria para escribir esa letra cuando recibe ese byte.
Aquí podemos notar una importante diferencia entre dos tipos de impresoras.

a) MATRICIAL O DE AGUJAS
b) MARGARITA O DAISY WHEEL

En el primer caso la información necesaria es la "forma" de la letra y en el segundo, el "lugar" donde se encuentra la letra. Veamos más detalladamente cada uno de estos tipos.

IMPRESORA MATRICIAL O DE AGUJA

Es la impresora más usada en computación, ya que reune características de velocidad y flexibilidad verdaderamente notables.



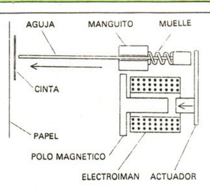


Figura 1

Figura 2

El sistema de impresión por aguja se basa en una cabeza de impresión formada generalmente por siete u ocho bobinas (electroiman) dispuestas en forma circular como se ve en la figura 1.

Cada bobina acciona una chapita (actuador) que actiua como un martillo sobre las agujas. La reposición de las agujas es realizada por la acción de un resorte de compresión, en cada una de ellas. Las agujas si bien están dispuestas en forma circular en la base del cono que forma la cabeza, son guiadas para quedar finalmente formando una sola línea en la punta del cono. Si con un impulso eléctrico se disparan todas las agujas y éstas a su vez pegan en una cinta entintada delante de un pepel, quedará impresa una raya vertical formado por 7 u 8 puntos alineados.

El funcionamiento es sumamente sencillo; se trata de dibujar las letras contando solamente con esta línea de agujas. El movimiento de esta línea a través del renglón produce una matríz de puntos. Si las agujas de la cabeza imprimieran siempre, producirían una matríz de puntos que a cierta distancia se vería como una raya gruesa. Por lo tanto para formar una letra lo único que debe hacer la lógica de la impresora es disparar las agujas en el momento correcto. Como se ve en la figura 2, para imprimir la letra "A" primero se disparan las agujas (PIN) 5,6 y 7 simultáneamente y luego solo la 4, y así sucesivamente hasta formar la letra por completo. Evidentemente este sistema permite dibujar cualquier tipo de letra, caracteres o símbolos especiales, siempre que la forma de ese caracter sea almacenada por la memoria ROM que tiene la impresora.

Otra importante ventaja es la velocidad con que imprimen, que como veremos en la próxima nota es mucho más rápida que la impresora tipo margarita.

INGENIERO NINO MORENO

SACANDOLE JUGO A LA CZ y

CZ 1000/1500
TK83/85

El microprocesador Z 80 es el cerebro de nuestras computadoras Sinclair. Sin embargo, no es muy listo por sí solo dado que no es capaz de entender sentencias como PRINT o LET. Cuando usamos estos comandos, no estamos contándole directamente al Z 80 qué es lo que queremos que ejecute, sino que son analizados primero por el INTERPRETE. Este intérprete configura por sí solo un programa del computador y está alojado entre las direcciones 0 y 8191 de la memoria ROM. Este programa está escrito en el llamado "lenguaje máquina" o "código máquina", lenguaje que el Z 80 puede entender directamente. Este lenguaje consiste en números entre 0 y 255. La tabla de estos números y sus mnemónicos las encontraremos en el manual BASIC. Los mnemónicos son etiquetas asignadas a cada código de máquina y describen en forma abreviada cuál es la función de cada código.

Podemos escribir programas en código de máquina, pero no es una tarea fácil.

Las principales ventajas de escribir programas en código máquina radican en que éstos se ejecutan a muy alta velocidad, no toman mucha memoria y son algunas veces más flexibles que el BASIC.

Los tres comandos BASIC que tratan con el código máquina son: PEEK, POKE y USR. PEEK n, nos permite examinar qué codigo se encuentra almacenado en la dirección de memoria n y POKE m, n coloca el código n en la dirección de memoria m. La rutina USR llama al Z 80 a ejecutar las instrucciones en código máquina almacenadas en una dirección particular.

Antes de usar el comando POKE para colocar números en memoria, debemos reservar el área para ellos. El manual BASIC explica en detalle cómo hacer esto.

Presentamos aquí un listado con cuatro rutinas en código de máquina que imitan funciones que son standard en computadores más costosos, y "agrandan" los 8 K ROM agregándole nuevos comandos:



Figura 1.
Sumario de nuevas rutinas

COMANDOS	DESCRIPCION
10 RAND USR IV	Convierte cualquier caracter de pantalla en su inverso.
10 PRINT USR FM	Muestra cuántos bytes libres quedan en memoria.
10 REM 10.2, SIN A	Estas son dos sentencias pseudo-DATA.
500 REM ABE, IKE	Pueden estar en cualquier sitio del programa. Cada entrada pueden ser números o letras seguidos de coma.
10 GOTO Z	Genera el listado del programa en inverso.
10 RAND USR RS	Ejecuta el comando RESTORE. Este debe ser usado antes de que cualquier DATA sea leída. Puede usarse para leer una misma data muchas veces.
10 LET A$ = " " 20 RAND USR RD 30 LET A = VAL A$	Hace el READ de la próxima entrada DATA. Omitir la línea 30 si sólo se leerán "sting" de caracteres.

El listado 1, es el programa cargador de los listados de código de máquina de las figuras 2 a 5.

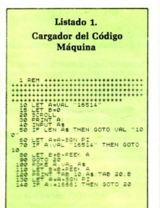
Podemos verificar las direcciones y el checksum mostrado por el programa con las direcciones y checksum de los listados. Si se comete un error, simplemente pulsamos ENTER para borrar la última entrada, y continuar normalmente con la carga.

Los usuarios de 1 K deberán usar CONT para seguir, cuando se produzca la detención del programa por overflow de pantalla.

Luego de tipear el último número, el programa se detendrá.

Tendremos ahora que borrar todas las líneas menos la primera, tipeando para esto el número de línea y luego ENTER.

Con la línea 1 en memoria, tipee y corra (RUN) el listado número 2. Este programa (listado 2) prepara para hacer SAVE a la cinta, por lo tanto debemos preparar el grabador para que reciba el SAVE. Antes de SAVE, tendremos que pulsar ENTER. La pantalla parpadeará un momento y luego comenzará con el SAVE. Ahora el código de máquina está ubicado en la parte superior de memoria, cualquiera sea el tamaño que estemos usando.

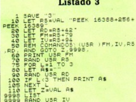


Listado 1.
Cargador del Código Máquina

```
   1 REM *************************
********************************
********************************
********************************
***************************
  10 LET A=VAL "16514"
  15 LET B=0
  20 SCROLL
  30 PRINT A;
  40 INPUT A$
  50 IF LEN A$ THEN GOTO VAL "10
0"
  60 LET A=A-SGN PI
  70 IF A<VAL "16514" THEN GOTO
10
  80 LET B=B-PEEK A
  90 GOTO 20
 100 POKE A,VAL A$
 110 LET B=B+PEEK A
 120 PRINT TAB 10;A$;TAB 20;B
 130 LET A=A+SGN PI
 140 IF A<=16661 THEN GOTO 20
```

Listado 2.

```
  10 SAVE "2"
  20 LET A=VAL "PEEK 16388+256+P
EEK 16389"
  30 INPUT B$
  40 POKE 16515,129
  50 LET A=A-120
  60 LET B=A+3
  70 LET C=VAL "16550"
  80 GOSUB VAL "130"
  90 LET B=A+29
 100 LET C=VAL "16566"
 110 GOSUB 130
 120 PRINT USR 16634
 130 LET E=VAL "INT (B/256)"
 140 LET D=B-E+256
 150 POKE C,D
 160 POKE C+SGN PI,E
 170 RETURN
```



Listado 3

```
   1 SAVE "3"
  10 LET RS=VAL "PEEK 16388+256+
PEEK 16389"
  20 LET RD=RS+42'
  30 LET FM=RS+79
  40 LET IV=RS+94
  50 REM COMANDOS:(USR )FM,IV,RS
,RD, AND GOTO Z,9998,
  60 PRINT USR FM
  70 RAND USR RS
  75 FOR L=1 TO 3
  80 LET A$=""
  90 RAND USR RD
 100 IF L<3 THEN PRINT A$
 105 NEXT L
 110 LET Z=VAL A$
9998 LIST
9999 RAND USR IV
```

Figura 2.
Rutinas Restore y Data

Listado Assembler		Direcc	Cod Maq			Checksum
RESTORE	LD HL,16634	16514	33	250	64	347
FIND	LD A, CODE "REM"	16517	62	234		643
	LD BC,65536	16519	1	255	255	1154
	CPIR	16522	237	177		1568
	LD BC,5	16524	1	5	0	1574
	AND A	16527	167			1741
	SBL HL,BC	16528	237	66		2044
	LD A,118	16530	62	118		2224
	CP (HL)	16532	190			2414
	JR Z,BELOW	16533	40	5		2459
	ADD HL,BC	16535	9			2468
STASH	LD (16507),HL	16536	34	123	64	2689
	RET	16539	201			2890
BELOW	ADD HL,BC	16540	9			2899
	JR FIND	16541	24	230		3153
READ CARACTER	LD HL, L(16507)	16543	42	123	64	3382
	LD A ,118	16546	62	118		3562
	CP (HL)	16548	190			3752
	CALL Z,FIND	16549	204	33	64	4053
	LD A, (HL)	16552	126			4179
	INC HL	16553	35			4214
	JR STASH	16554	24	236		4474
READ STRING	LD HL,(E-LINE)	16556	42	20	64	4600
	DEC HL	16559	43			4643
	PUSH HL	16560	229			4872
	DEC HL	16561	43			4915
	DEC HL	16562	43			4958
	EX (SP),HL	16563	227			5185
LOOP	PUSH HL	16564	229			5414
	CALL READ CARACTER	16565	205	159	64	5842
	POP HL	16568	225			6067
	LD B,CODE","	16569	6	26		6099
	CP B	16571	184			6283
C.DONE	JR Z,DONE	16572	40	13		6336
	LD(HL),A	16574	119			6455
	INC HL	16575	35			6490
INC LEN$	EX (SP),HL	16576	227			6717
	INC (HL)	16577	52			6769
	JR NZ,NO CARRY	16578	32	3		6804
	INC HL	16580	35			6839
	INC(HL)	16581	52			6891
	DEC HL	16582	43			6934
NO CARRY	EX (SP),HL	16583	227			7161
DO MORE	AND A	16584	167			7328
	JR NC,LOOP	16585	48	233		7609
DONE	EX (SP),HL	16587	227			7836
	POP HL	16588	225			8061
	NOP	16589	0			8061
	JP ROM	16590	195	157	20	8433

SACANDOLE JUGO A LA CZ Y TK

Figura 3.
Rutina de Bytes Libres

Listado Assembler		Direcc	Cod Maq			Checksum
FREE MEM	LD HL, (STKEND)	16593	42	28	64	8567
	LD B, H	16596	68			8635
	LD C, L	16597	77			8712
	LD HL, 0	16598	33	0	0	8745
	ADD HL, SP	16601	57			8802
	CP A	16602	191			8993
	SBC HL, BC	16603	237	66		9296
	LD B, H	16605	68			9364
	LD C, L	16606	77			9441
	RET	16607	201			9642

Figura 4.
Rutina de Inversión

Listado Assembler		Direcc	Cod Maq			Checksum
IV	LD HL, (D-FILE)	16608	42	12	64	9760
	LD B, H	16611	68			9828
	LD C, L.	16612	77			9905
	LD D, 59	16613	22	59		9986
	SLA D	16615	203	34		10223
LOOP	LD HL, (VARS)	16617	42	16	64	10345
	LD A, (BC)	16620	10			10355
	CP D	16621	186			10541
	JR Z, AGAIN	16622	40	3		10584
	ADD A, 128	16624	198	128		10910
	LD (BC), A	16626	2			10912
AGAIN	INC BC	16627	3			10915
	CP A	16628	191			11106
	ABC HL, BC	16629	237	66		11409
	JR NZ, LOOP	16631	32	240		11681
	RET	16633	201			11882

Aplicaciones prácticas:

El listado 3 demuestra el uso de varios de los comandos. También declara las variables RS, RD, IV, FM y Z. En este listado Z es leída en una línea REM.
Debemos poner a grabar el grabador antes de hacer RUN al listado 3, porque esto pone en SAVE al computador. Luego de esto se deberán borrar todas las líneas excepto la última, de lo contrario el computador se pondrá en SAVE nuevamente. No deben usarse NEW o CLEAR, porque así las variables pueden pasar a la cinta.

El procedimiento para cargar los nuevos comandos es muy sencillo, simplemente LOAD listing 2. A continuación, cualquiera de las rutinas con comandos extendidos.
Precaución: Estos listados funcionarán sólo con la misma configuración de memoria que tenía el computador antes de ser SAVEados a la cinta.

Como algunas aplicaciones, podemos por ejemplo, ver el caso en que se tengan que imprimir muchos títulos con poca memoria, probablemente el computador indique un error "4" indicando un overflow de pantalla. Esto puede ser prevenido con la siguiente línea:

```
50 IF USR FM • 100 THEN CLS
```

Esta línea evita el tipear "CONT" para que el programa arranque nuevamente. Pero, la pantalla se borrará tal vez antes de que se pueda leer toda la línea, para prevenir esto, agregaremos unas líneas más:

```
50 IF USR FM • 100 THEN
PAUSE 40000
60 IF USR FM • 100THEN CLS
```

Esto detendrá el programa dando



Figura 5.
Rutina de Re-ubicación

Listado Assembler		Direcc	Cod Maq			Checksum
Relocate	LD HL,(RAMTOP)	16634	42	4	64	11992
	LD BC,-120	16637	1	136	255	12384
	ADD HL,BC	16640	9			12393
	PUSH HL	16641	229			12622
	PUSH HL	16642	229			12851
	EXX	16643	217			13068
	POP BC	16644	193			13261
	EXX	16645	217			13478
	LD HL,16514	16646	33	130	64	13705
	POP DE	16649	209			13914
	LD BC,120	16650	1	120	0	14035
	LDIR	16653	237	176		14448
	EXX	16655	217			14665
	DEC BC	16656	11			14676
	OUT 253,A	16657	211	253		15140
	JP NEW	16659	195	203	3	15541

tiempo para leer con comodidad, pulsando cualquier tecla, continuará limpiando primero la pantalla. La versión que se elija, dependerá de la aplicación a la que se la destine. Se puede utilizar la rutina de inversión para crear interesantes efectos visuales, o por ejemplo, hacer más descansada la lectura desde la pantalla, etc.

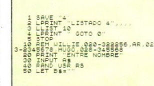
Si hay que recordar números, listas, etc., entonces las rutinas de RESTORE y READ son ideales para esto. Se pueden usar las sentencias REM para guardar importantes números de teléfono, fechas, cumpleaños, etc. Pueden ser útiles en inventarios, coordenadas de pantallas para PLOT, direcciones de código de máquina o palabras. El listado 4 es un ejemplo de directorio telefónico por computador, y corre con 1 K.

El listado 5 graficará cualquier función que le definamos. Primero: debemos entrar en la función en términos de X. Por ejemplo: si buscamos graficar F (x) =(4X + 3X +2) entonces haremos el input con:

4 * X * X + 3 * X + 2

Luego necesitamos fijar los límites de X superior e inferior si plotteamos la curva "sen X66, probablemente elijamos como límites cero (0) y dos pi. Los límites verticales son calculados automáticamente. Almacenaremos en sentencias REM las funciones a graficar junto con sus dos parámetros superior e inferior.

Listado 4.



```
   1 SAVE "4"
   2 LPRINT "LISTADO 4",,,,
   3 LLIST 10
   4 LPRINT " GOTO 0"
   5 STOP
  10 REM WILLIE,020-322256,AR,02
3-2345678,HUGO,028-345568
  20 PRINT "ENTRE NOMBRE"
  30 INPUT A$
  40 RAND USR RS
  50 LET B$=""
  60 RAND USR RD
  70 IF B$="OUT OF DATA" THEN GO
TO VAL "20"
  80 LET C$=""
  90 RAND USR RD
 100 IF B$<>A$ THEN GOTO VAL "50
"
 110 PRINT C$
 120 GOTO VAL "20"
 130 REM OUT OF DATA
```

Listado 5.

```
   1 SAVE "5"
   2 LPRINT "LISTADO 5",,,,
   3 LLIST 10
   4 LPRINT " GOTO 0"
   5 STOP
  10 LET F=SGN PI
  20 GOSUB CODE " COPY "
  25 IF F THEN PRINT "FUNCION"
  30 INPUT A$
  35 IF F THEN PRINT "LIMITE INF
."
  40 INPUT X
  45 IF F THEN PRINT "LIMITE SUP
."
  50 INPUT A
  60 LET K=X
  70 FAST
  80 IF F THEN CLS
  85 LET F=NOT PI
  90 LET DX=(A-X)/CODE "Z"
 100 LET H=VAL A$
 110 LET L=H
 120 FOR I=NOT PI TO CODE "Z"
 130 IF H<VAL A$ THEN LET H=VAL
A$
 140 IF L>VAL A$ THEN LET L=VAL
A$
 150 LET X=X+DX
 160 NEXT I
 170 LET X=K
 180 SLOW
 190 FOR I=NOT PI TO CODE "Z"
 200 PLOT I,VAL "43"*(VAL A$-L)/
(H-L)
 210 LET X=X+DX
 220 NEXT I
 230 GOSUB VAL "400"
 240 GOTO VAL "30"
 300 RAND USR RS
 310 FOR A=VAL "-2" TO VAL "10"
 320 FOR B=NOT PI TO VAL "24" ST
EP VAL "8"
 330 LET A$=""
 340 IF B OR A<NOT PI THEN RAND
USR RD
 350 IF NOT B AND A>-SGN PI THEN
 LET A$=STR$ A
 360 PRINT TAB B;A$
 370 NEXT B
 380 NEXT A
 390 REM ***,ELIJA ,OPCION,***,O
PCION,FN,RANGO,,INPUT, SU ,    ,F
N,
 400 INPUT B
 420 IF NOT B THEN RETURN
 430 RAND USR RS
 440 FOR A=SGN PI TO VAL "3*B+8"
 450 LET A$=""
 460 RAND USR RD
 470 NEXT A
 480 FOR A=-SGN PI TO SGN PI
 491 LET B$=""
 500 RAND USR RD
 510 IF A<NOT PI THEN LET A$=B$
 520 IF NOT A THEN LET X=VAL B$
 530 NEXT A
 540 LET A=VAL B$
 550 GOTO VAL "60"
1000 REM SIN X,0,2*PI,COS X,0,2*
PI,TAN X,-1,1,SIN X+SIN (2*X)+SI
N (3*X),0,4*PI,ASN X,-1,1,LN X,.
1,2.7,EXP X,-1,1,ABS (X-INT X-.5
),0,3,1/X,.1,2,EXP -(X*X),-2,2,
```

CONOCIENDO EL BUZON

ING. PEDRO E. COLLA

En general, el T82068 soporta en su conector para cartridge dispositivos exteriores de prácticamente cualquier tipo, pues están allí las principales líneas del bus. Ello no es contradictorio con que primariamente el mismo este orientado a la utilización de software en ROM (memoria de lectura solamente).
La memoria del computador tiene una distribución tal como se especifica en la Figura 1. Nótese que hay una zona en la cual se superponen la memoria ROM principal y la llamada de "Extensión". En realidad no es que ambos bancos puedan funcionar o estar visibles al mismo tiempo, sino que se realiza la conmutación entre ambos en la medida que se quiera ejecutar lo que contiene uno u el otro. En una anterior entrega en esta publicación se dio un ejemplo de tal situación.

Independientemente del contenido original, la memoria se divide en secciones cuyo tamaño es de 8 kbytes denominadas "cuartiles" o "chunks", teniendo el hardware la posibilidad de eliminar mediante instrucciones de software adecuadas una o más de esas zonas. Esto facilita enormemente el agregado externo de software en reemplazo del interno pues es posible anular este último mediante el apagado del "chunk" correspondiente y por supuesto su reemplazo con el que nos resulte conveniente.

De esta menera es posible colocar en el buzón software en ROM que reemplace parte o la totalidad de ROM del computador, sin tener que recurrir a engorrosos artificios circuitales, como era necesario hacer por ejemplo en el computador TS1000.

En realidad, las cosas son más fáciles aun pues el computador como parte de su rutina de encendido (cuando la pantalla se vuelve negra) realiza por sí mismo una lectura en el conector de cartridge, y si detecta que allí hay algo, toma los recaudos para utilizarlo y que el resto del computador no interfiera con ello.



Se prevee que puedan existir dos tipos de software en cartridge, lo que se denomina LROS y su contrapartida el AROS.
El primero de ellos (LROS) está pensando para software en lenguaje de máquina que reemplace al (los) ROM original del computador para proveer al mismo de una modalidad de funcionamiento enteramente diferente. Claros ejemplos de uso de esta alternativa serían hacer que el principal lenguaje fuera FORTH y no BASIC, o aprovechando la existencia del procesador Z80 el lograr que se descarte el modo de funcionamiento del Sinclair para transformarlo en un sistema con diskettes bajo CP/M.

En cambio en el caso del AROS se prevee la utilización de software que necesite del ROM Sinclair, es decir de aplicaciones o juegos. Además en el caso del AROS el contenido del cartridge puede estar tanto en lenguaje de máquina como en BASIC y por cierto no elimina al ROM principal.
Físicamente un cartridge es una plaqueta de circuito impreso conteniendo uno o más chips de memoria ROM o EPROM. El tipo de memoria a utilizar está fundamentalmente dado por el volúmen de producción que se prevea. En el caso de software comercial, las memorias tipo ROM son posiblemente la solución de menor costo mientras que para proyectos esencialmente unitarios, es infinitamente más practica la utilización de memorias EPROM (memorias de

Una de las características que distinguen al computador Timex-Sinclair 2068 de su similar europeo, el SPECTRUM, es el conector para cartridge, más vulgarmente conocido como buzón. Es posible agregarlo a esta máquina, pero en forma opcional.

Lectura Solamente y Borrables). Las primeras posiciones de la misma contienen la ifnormación necesaria para que al realizar el Proceso de inicialización el computador pueda conocer si el contenido de la misma es de tipo LROS o AROS y cuáles son los chunks de memoria que resultan afectados.
En el caso del LROS, que es el que nos interesa, son los primeros 5 bytes los que contienen tal información y su estructura es la siguiente:

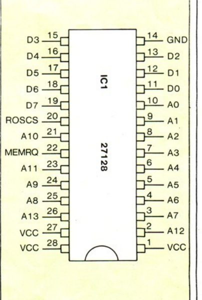
0000	No es utilizada
0001	Contiene el valor 01 indicando LROS.
0002/0003	Dirección de primera instrucción a ejecutar
0004	Especificación de chunks a anular.

Nótese que si no hay cartridge conectado el computador, "ve" solamente direcciones conteniendo el valor 255 y por lo tanto al leer la posición 0001 se da por notificado que no hay ningún LROS para ejecutar (y tampoco ningún AROS cuyo valor para posición 0001 es 02) y continúa operando normalmente.
Una vez que el computador se notifica que existe una sección de LROS, utiliza el quinto byte para saber que secciones de memoria debe reemplazar con el. Las posiciones segunda y tercera le dan la información acerca de a que dirección de memoria se deberá transferir la ejecución del programa luego de haber finalizado la inicialización.

FIGURA 1
MAPA DE MEMORIA DEL COMPUTADOR TS2068

DIRECCIONES	CONTENIDO		CHUNK
00000 A 08191	ROM	ROM	CHUNK 0
08192 A 16383	PRINCIPAL	EXTEND.	CHUNK 1
16384 A 24575			CHUNK 2
24576 A 32767			CHUNK 3
32768 A 40959			CHUNK 4
40960 A 49151			CHUNK 5
49152 A 57313	RAM		CHUNK 6
57314 A 65535	PRINCIPAL		CHUNK 7

Figura 2
Conexiones del EPROM al conector del Cartridge



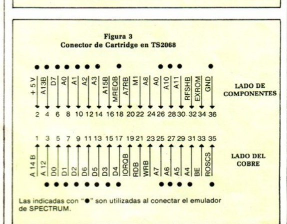


Figura 3
Conector de Cartridge en TS2068

Las indicadas con "●" son utilizadas al conectar el emulador de SPECTRUM.

CONVERSION DE LA TS 2068

La especificación de que chunks son anulados y cuáles no, se logra numerando cada uno de 0 a 7 (8 chunks de 8 kbytes cada uno suman 64 Kbytes, la totalidad de memoria), cada bit de esta posición refiere a un chunk específico, siendo el bit 0 el que controla el segmento que va entre las direcciones 0 y 8191, el bit 2 al que transcurre entre 8192 y 16383 y así sucesivamente. De tal manera que si el bit asociado tiene un valor "1", el chunk correspondiente no debe ser anulado, mientras que si su valor es "0" se debe proceder a la desconexión del mismo. Para clarificar esto con un ejemplo supongamos que queremos reemplazar con el contenido de un cartridge la totalidad del ROM del computador, el que vimos en Fig. 1 que se extiende entre las posiciones 0 y 16383. Los primeros cinco bytes del cartridge deberán entonces contener.

0000 000
0001 001
0002 000
0003 019
0004 252 1111 1100 anula los chunks 0 y 1.

Es interesante apuntar que la mayoría del software ideado para computadores SPECTRUM no es compatible para su ejecución en máquinas TS2068, y por lo tanto aprovechando la similitud del hardware entre ambas es factible hacer que el computdor TS2068 "emule" un SPECTRUM colocando el ROM de este último en reemplazo del original del primero.

Con este cambio el TS2068 sin duda pierde potencia, pues no puede utilizar en forma sencilla los joysticks, el sintetizador musical y algunas instrucciones especiales del BASIC (ON ERR, STICK, RESET, SOUND, etc.) Pero se gana en el acceso al constantemente renovado software para SPECTRUM que en calidad y cantidad supera el disponible para máquinas TS2068. Para lograr ello deberemos reproducir en una EPROM el contenido del ROM de un computador SPECTRUM cualquiera y modificar las posiciones de memoria necesarias para cumplir los requisitos de un LROS.

Tendremos entonces que concurrir a una de las numerosas casas especializadas en grabación de EPROM portando el chip "vacío" junto con una ROM de cualquier computador SPECTRUM, posiblemente prestada por un amigo, para que se obtenga un "duplicado" de la misma. Normalmente con un bajo o nulo cargo adicional es posible que antes de asentar la copia se realicen las moficaciones que se consignan en la presente, para que dicho chip pueda funcionar como LROS.

En las Figuras 2 y 3 se brindan el esquema de conexiones del circuito integrado y el diagrama del zócalo de cartridge del computador. Nótese que debido a la extrema sencillez del conexionado, no se justifica el armado de una plaqueta de circuito impreso, siendo posible el armado del circuito con técnicas de soldado punto a punto o "wire-wrapping".

Como integrado EPROM se utiliza un 27128 cuya capacidad es de 16K bytes X 8 bits, por lo que la totalidad del ROM SPECTRUM puede ser albergada en un sólo integrado. En caso de dificultades en la obtención del citado elemento electrónico, nada impide lograr el mismo propósito mediante dos chips 2764 (8 Kbytes X 8 bits) o cuatro chips 2732 (4Kbytes X 8 bits) de más fácil obtención, aunque en ambos casos se complicará ostensiblemente el aspecto mecánico del montaje.

Para que el integrado grabado funcione correctamente, debe tenerse en cuenta que las primeras posiciones no podrán coincidir exactamente con el ROM SPECTRUM, pues deben contener los datos necesarios para que durante la inicialización se disponga de la información para utilizarlo correctamente. Para ello se aprovecha un hueco de algunos bytes en la posición 19 del ROM SPECTRUM para transferir allí las instrucciones ubicadas originalmente en las primeras direcciones. Por ello se debe grabar en forma diferente las primeras 30 posiciones sobre el integrado de EPROM, siendo el contenido final de las mismas el expresado por la Figura 4. El resto de las posiciones son un fiel reflejo de las contenidas en el ROM SPECTRUM. Al encender el computador con el cartridge ocupado por el circuito que nos ocupa, y si hemos hecho bien las cosas, el computador debería "inicializarse" como lo hace normalmente, llegando a exhibir durante algo menos de un segundo la leyenda de "copyright" normal para a continuación volver a inicializarse, para aparecer al final de dicho proceso el mensaje de "copyright" del computador SPECTRUM; de allí en adelante a todos los efectos prácticos del computador dejará de comportarse como un TS2068 u será un SPECTRUM. Si ésto no ocurre significará que se ha cometido algún error en la grabación de la EPROM o en el conexionado del circuito para lo cual habrá que revisar cuidadosamente ambos puntos. En ningún caso se puede conectar o desconectar el cartridge con el computador encendido pues el mismo se puede dañar. Exceptuando esta última precaución, es improbable que un error en el circuito produzca daño alguno y mucho menos permanente.

FIGURA 4
POSICIONES INICIALES PARA EMULADOR DE SPECTRUM

DIRECCION	+00	+01	+02	+03	+04	+05	+06	+07	+08	+09
00000	255	001	019	000	252	255	255	255	042	093
00010	092	034	095	092	024	067	195	242	021	017
00020	255	255	024	019	042	093	092	126	205	125
00030 A 16383	IGUAL A ROM SPECTRUM									



2[do]
CONCURSO TRIMESTRAL

PATROCINA

SANWA



1er. Premio: Una Consola COMMODORE 64
2do. Premio: Una Consola SPECTRUM
3er. Premio: Dos Pasajes a Bariloche
4to. Premio: Un Grabador para Computadora
5to. Premio: Una Impresora Alpha Com 32

Condiciones para participar en el certamen:

1: Los programas deberán ser originales e inéditos, pudiendo cubrir todas las áreas: educativos y de cálculo, uso comercial, entretenimiento, personal y utilitarios para programación, hasta 64 K. 2: El criterio de elección se basará en: originalidad de la idea, método de programación, efectos gráficos y/o sonoros, documentación, presentación y ahorro de memoria. 3: Se enviarán a K-64 grabados en un cassette y acompañados por el listado correspondiente con pantallas y explicacion sobre la utilidad y manejo del programa. 4: Puede remitirse más de un programa por cassette, en lo posible grabados dos veces, para mayor seguridad. 5: El concurso K-64 se realizará trimestralmente. El cierre de la recepción de los trabajos será el 15/9/85. 6: K-64 se reserva el derecho de publicación de los programas enviados (como asimismo de la devolución del material recibido).

El cassette deberá ser enviado con su caja y con los datos del programa y del autor, como así también de la computadora para la cual está destinado.

SELECCION MENSUAL

Mensualmente se seleccionarán 50 Programas, los que se harán acreedores a los siguientes premios: Calculadoras, Máquinas de fotos, cassettes con programas, cassettes vírgenes, Becas para Cursos, etc.

Los Programas seleccionados continúan en Concurso para la gran final Trimestral.

Presentando este cupón obtendrán un 10% de descuento de las compras que realicen en SANWA S.A. y un 50% para los cursos en EPI.

RETIRO DE CUPONES: K-64: Cerrito 1320 - SANWA: Av. Corrientes 2198, Florida 683 - EPI: Suipacha 946 - 1er. piso, Viamonte 1479 - 8° Piso "B", Florida 683, Av. Corrientes 2198 - Radio del Plata: Av. Santa Fe 2043.

AUSPICIAN



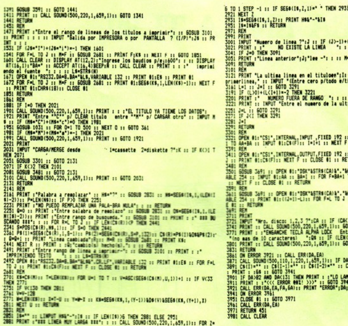


EDITOR DE TEXTOS

TI 99/4A




```
2 CALL CLEAR :: FOR A=1 TO 10 :: CALL COLOR(A,6,16):: NEXT A :: CALL SCREEN(16):
: CALL COLOR(9,6,16):: CALL CHAR(33,"00000000F0F0F0F0"):: CALL CHAR(34,"F0F0F0F0
")
3 CALL CHAR(35,"FFFFFFFFF0F0F0F0"):: CALL CHAR(36,"F0F0F0F0F0F0F0F0"):: CALL CHA
R(37,"FFFFFFFFFFFFFFFF"):: CALL CHAR(38,"FF00FF00FF00FF00")
4 CALL CHAR(40,"F000F000F000F000"):: CALL VCHAR(4,3,37,2):: CALL VCHAR(4,6,37,2)
:: CALL VCHAR(4,9,37,2):: CALL HCHAR(5,4,33):: CALL HCHAR(4,4,34)
5 CALL HCHAR(3,3,35):: CALL HCHAR(3,4,34):: CALL HCHAR(5,5,33):: CALL HCHAR(3,6,
35):: CALL HCHAR(3,7,36):: CALL HCHAR(4,7,34)
6 CALL HCHAR(5,8,33):: CALL HCHAR(3,9,36):: CALL HCHAR(5,11,33):: CALL HCHAR(2,9
,34):: CALL HCHAR(24,1,38,30):: CALL VCHAR(1,30,40,24)
7 A$="EMPRESA PARA INFORMATICA" :: B=8 :: C=3 :: GOSUB 10 :: A$="PROGRAMAS PARA
EL" :: B=12 :: C=3 :: GOSUB 10 :: A$="MICROCOMPUTADOR TI-99/4A" :: B=14 :: C=3 :
: GOSUB 10
8 A$="FABRICADO Y DISTRIBUIDO  EN" :: B=19 :: C=3 :: GOSUB 10 :: A$="LA REPUBLIC
A  ARGENTINA POR" :: B=20 :: C=3 :: GOSUB 10 :: A$="'SDT'INTELIGENCIA ARGENTINA"
 :: B=22 :: C=3
9 GOSUB 10 :: FOR D=1 TO 700 :: NEXT D :: GOTO 11
10 FOR D=1 TO LEN(A$):: B$=SEG$(A$,D,1):: E=ASC(B$):: CALL HCHAR(B,C-1+D,E):: NE
XT D :: RETURN
11 ON WARNING NEXT :: CALL CHARSET :: CALL CLEAR
12 CALL SCREEN(12):: DISPLAY AT(12,2):"    EDITOR DE TEXTOS" :: FOR TT=1 TO 500
:: NEXT TT
13 DISPLAY AT(2,2)BEEP ERASE ALL:"ESTE PROGRAMA LE PERMITIRA  OPERAR EN PANTALLA
 O POR IM-PRESORA CON UN VERDADERO EDITOR DE TEXTOS."
14 DISPLAY AT(7,2):"LOS TEXTOS PUEDEN QUEDAR    ALMACENADOS EN CASSETTE O ENDISK
ETTE.   "
15 DISPLAY AT(11,2):"NO REDUNDAREMOS EN MAYORES  EXPLICACIONES, PUESTO QUE   PEN
SAMOS QUE EL PROGRAMA ES AUTOCONTENIDO EN SUS APLICA-"
16 DISPLAY AT(16,2):"CIONES Y MODOS DE OPERACION,SOLO ESPERAMOS QUE DISFRUTE DE
ESTE UTILITARIO REALIZADOPARA SU TI-99/4A."
18 FOR EE=1 TO 200 :: NEXT EE
20 DISPLAY AT(24,1):"OPRIMA LA TECLA ENTER" :: ACCEPT AT(24,24)BEEP:QQQ$
21 CALL CLEAR
81 OPTION BASE 1 :: DIM C$(200),D$(31)
141 DATA NUL,SOH,STX,ETX,EOT,ENQ,ACK,BEL,BS,HT,LF,VT,FF,CR,SO,SI,DLE,DC1,DC2,DC3
,DC4
151 DATA NAK,SYN,ETB,CAN,EM,SUB,ESC,FS,GS,RS,US
171 FOR F=1 TO 31 :: READ D$(F):: NEXT F :: E$=CHR$(18)&CHR$(20)&CHR$(27)&CHR$(7
0)&CHR$(27)&CHR$(72)&CHR$(13)&CHR$(27)&"5" :: F$="$" :: G=80 :: CALL CLEAR :: CA
LL SCREEN(16)
331 G$="LO LAMENTO,MEMORIA LLENA  "
361 CALL CLEAR
371 DISPLAY AT(3,1):"      EDITOR  DE  TEXTOS          funciones  comando" :: DI
SPLAY AT(7,1):"1-Borra  2-Agrega 3-Reemplaza  linea    linea    linea "
391 DISPLAY AT(10,1):"4-Borra  5-Crea   6-Graba     titulo   titulo   titulo" ::
 DISPLAY AT(13,1):"7-Lista  8-Carga  9-Reemplaza  titulo   titulo   palabras"
411 DISPLAY AT(16,1):"0-Para ver los comandos"
421 DISPLAY AT(20,1):"CUANTOS CARACTERES POR LINEAQUIERE  ?";G :: ACCEPT AT(21,1
1)VALIDATE(DIGIT)SIZE(-3):G :: IF (G<1)OR(G>254)THEN 421
451 PRINT :: ON ERROR 3861 :: CALL SCREEN(16):: INPUT "Accion ? ":H
471 IF H=-1 THEN 3981 :: H=H+1
481 IF (H<1)+(H>10)=-1 THEN 491 ELSE 511
491 CALL CLEAR :: DISPLAY AT(1,1):"<<ERROR DE SELECCION>>" :: GOTO 371
511 ON H GOSUB 361,541,691,881,1031,1101,1251,1451,1861,2141
521 IF H=9 THEN CALL CLEAR :: GOTO 361
531 GOTO 451
541 REM
561 IF I>0 THEN 601
581 PRINT :"    NO HAY TITULO   ": :: GOTO 681
601 GOSUB 2971 :: IF J=0 THEN 681
621 I=I-1 :: FOR F=J TO I :: C$(F)=C$(F+1):: NEXT F :: C$(I+1)="" :: PRINT :"Bor
rado,titulo renumerado":
681 RETURN
691 REM
711 CALL SCREEN(16):: IF I>0 THEN 751
731 PRINT :"   NO HAY TITULO   ": :: GOTO 871
751 PRINT :"Agregue despues" :: GOSUB 2971 :: IF J=0 THEN 871
781 PRINT :"Entre nueva linea-" :: H$=STR$(J):: GOSUB 2831 :: I=I+1 :: FOR F=I T
O J+1 STEP -1 :: C$(F)=C$(F-1):: NEXT F :: C$(J)=I$ :: PRINT :"Linea agregada,ti
tulo renumerado": :
871 RETURN
881 REM
901 IF I>0 THEN 941
921 PRINT :"   NO HAY TITULO   ": :: GOTO 1021
941 PRINT :"Reemplace" :: GOSUB 2971 :: IF J=0 THEN 1021
971 PRINT :"Entre linea de reemplazo-": :: H$=STR$(J):: GOSUB 2831 :: C$(J)=I$ ::
 PRINT :"Linea reemplazada": :
1021 RETURN
1031 REM
1041 FOR F=1 TO I :: C$(F)="" :: NEXT F :: I=0 :: PRINT : :"  AREA DE MEMORIA LI
MPIA": : :: RETURN
1101 REM
1121 CALL SCREEN(16):: CALL CLEAR :: PRINT : : :"--- Lista para tipeo ---
Entre ^^^ para fin)": :
1171 H$=STR$(I+1):: GOSUB 2831 :: IF LEN(I$)<>4 THEN 1211
1201 IF SEG$(I$,1,2)="^^" THEN 1241
1211 I=I+1 :: C$(I)=I$ :: GOTO 1171
1241 RETURN
1251 REM
1271 IF I>0 THEN 1311
1291 PRINT : :"   NO HAY TITULO   ": : :: GOTO 1441
1311 PRINT : :"Entre rangos titulo a grabar": : :: GOSUB 3101 :: PRINT
1341 INPUT "Graba en                         1=cassette  2=diskette:":K :: IF K<>1 T
HEN 1381
1361 GOSUB 3391 :: GOTO 1441
1381 IF K<>2 THEN 1411
```



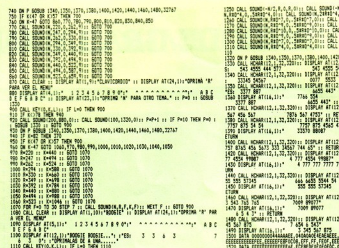
```
1391 GOSUB 3591 :: GOTO 1441
1411 PRINT :: CALL SOUND(500,220,1,659,1):: GOTO 1341
1441 RETURN
1451 REM
1471 PRINT :"Entre el rango de lineas de los titulos a imprimir": :: GOSUB 3101
:: PRINT : : : :: INPUT "Salida por IMPRESORA o por  PANTALLA  ? (I/P)":J$ :: PR
INT : : :
1531 IF (J$="I")+(J$="i")=-1 THEN 1601
1541 FOR F=L TO J :: M=F :: GOSUB 2681 :: PRINT F;K$ :: NEXT F :: GOTO 1851
1601 CALL CLEAR :: DISPLAY AT(12,2):"Ingrese los baudios p/e;:600": : :: DISPLAY
 AT(16,1):"BA=" :: ACCEPT AT(16,4)BEEP:N :: CALL CLEAR :: PRINT : : :"   imprimi
endo el fuente  ": : : :: L$=STR$(N)
1671 OPEN #1:"RS232.DA=8.BA="&L$,VARIABLE 132 :: PRINT #1:E$ :: PRINT #1
1672 FOR F=L TO J :: M=F :: GOSUB 2681 :: PRINT #1:SEG$(K$,1,LEN(K$)-1):: NEXT F
 :: PRINT #1:CHR$(18):: CLOSE #1
1851 RETURN
1861 REM
1881 IF I=0 THEN 2021
1901 CALL SOUND(500,220,1,659,1):: PRINT : : :"EL TITULO YA TIENE LOS DATOS":
1921 PRINT "Entre ""C"" p/ CLEAR titulo   entre ""M"" p/ CARGAR otro" :: INPUT M
$ :: IF (M$="C")+(M$="c")=0 THEN 1981
1951 GOSUB 1031 :: FOR O=1 TO 500 :: NEXT O :: GOTO 361
1981 IF (M$="M")+(M$="m")=-1 THEN 2021
1991 CALL SOUND(500,220,1,659,1):: PRINT :: GOTO 1921
2021 PRINT
2031 INPUT "CARGA/MERGE desde          1=cassette  2=diskette ?":K :: IF K<>1 T
HEN 2071
2051 GOSUB 3301 :: GOTO 2131
2071 IF K<>2 THEN 2101
2081 GOSUB 3481 :: GOTO 2131
2101 CALL SOUND(500,220,1,659,1):: PRINT :: GOTO 2031
2131 RETURN
2141 REM
2161 PRINT :"Palabra a reemplazar" :: H$="?" :: GOSUB 2831 :: N$=SEG$(I$,1,(LEN(I
$)-2)):: P=LEN(N$):: IF P>0 THEN 2251
2231 PRINT :"NO PUEDO REMPLAZAR UNA PALA-BRA NULA": : :: RETURN
2251 Q=0 :: PRINT :"Entre palabra de reemplazo" :: GOSUB 2831 :: D$=SEG$(I$,1,(LE
N(I$)-2)):: PRINT :"Entre rango de busqueda." :: GOSUB 3101 :: PRINT : :" ### BU
SCANDO ###": : :: FOR R=L TO J :: IF LEN(C$(R))<P THEN 2441
2341 S=POS(C$(R),N$,1):: IF S=0 THEN 2441
2361 P$(1)=SEG$(C$(R),1,S-1):: P$(2)=SEG$(C$(R),S+P,132):: C$(R)=P$(1)&D$&P$(2):
: Q=Q+1 :: PRINT "Linea cambiada":R:: M=R :: GOSUB 2681 :: PRINT K$:
2441 NEXT R :: PRINT : :Q;"cambio(s) hecho(s).": : :: RETURN
2491 PRINT :"Entre rango de linea a imprimir-": :: GOSUB 3101 :: PRINT : :"
 IMPRIMIENDO TEXTO    ": : :: L$=STR$(N)
2492 OPEN #1:"RS232.DA=8.BA="&L$&".CR.LF",VARIABLE 132 :: PRINT #1:E$ :: FOR F=L
 TO J :: PRINT #1:C$(F):: NEXT F :: CLOSE #1 :: RETURN
2681 REM
2701 K$=C$(M):: T=LEN(K$):: FOR U=1 TO T :: V=ASC(SEG$(C$(M),U,1))+1 :: IF V<33
THEN 2771
2751 IF V<130 THEN 2811
2761 V=V-128
2771 W=LEN(K$):: X=T-U :: Y=W-X :: K$=SEG$(K$,1,(Y-1))&D$(V)&SEG$(K$,(Y+1),X)
2811 NEXT U :: RETURN
2831 REM
2851 I$="" :: LINPUT H$&"-":I$ :: IF LEN(I$)>6 THEN 2881 ELSE 2951
2881 PRINT :"### LINEA MUY LARGA ###": : :: CALL SOUND(500,220,1,659,1):: FOR Z=
6 TO 1 STEP -1 :: IF SEG$(I$,Z,1)=" " THEN 2931
2921 NEXT Z
2931 I$=SEG$(I$,1,Z):: PRINT H$&"-"&I$
2951 I$=I$&F$ :: RETURN
2971 REM
2991 PRINT
3001 INPUT "Numero de linea ?":J :: IF (J>-1)+(J<(I+1))=-2 THEN 3041
3021 PRINT : :"     NO EXISTE LA LINEA     ": : :: GOTO 3001
3041 IF J=0 THEN 3091
3051 PRINT :"Linea anterior";J;"lee -": : :: M=J :: GOSUB 2701 :: PRINT K$
3091 RETURN
3101 REM
3121 PRINT "La ultima linea en el tituloes";I: :: PRINT :"Entre el numero de la
primerlinea," :: INPUT "(Entre cero p/todo e/titulo):":L :: IF L>0 THEN 3191
3161 L=1 :: J=I :: GOTO 3291
3191 IF (L>0)+(L<I+1)=-2 THEN 3221
3201 PRINT :"   NUMERO FUERA DE RANGO  ": : :: GOTO 3121
3221 PRINT :: INPUT "Entre el numero de la ultimalinea:":J :: IF J>L THEN 3271
3251 J=L :: GOTO 3291
3271 IF J<I THEN 3291
3281 J=I
3291 RETURN
3301 REM
3321 OPEN #1:"CS1",INTERNAL,INPUT ,FIXED 192 :: INPUT #1:AA :: BA=I :: FOR F=BA+
1 TO AA+BA :: INPUT #1:C$(F):: I=I+1 :: NEXT F :: CLOSE #1 :: RETURN
3391 REM
3411 OPEN #1:"CS1",INTERNAL,OUTPUT,FIXED 192 :: PRINT #1:((J+1)-L):: FOR F=L TO
J :: PRINT #1:C$(F):: NEXT F :: CLOSE #1 :: RETURN
3481 REM
3501 GOSUB 3691 :: OPEN #1:"DSK"&STR$(CA)&"."&Q$,INPUT ,SEQUENTIAL,INTERNAL,VARI
ABLE 254 :: INPUT #1:AA :: BA=I :: FOR F=BA+1 TO AA+BA :: INPUT #1:C$(F):: I=I+1
 :: NEXT F :: CLOSE #1 :: RETURN
3591 REM
3611 GOSUB 3691 :: OPEN #1:"DSK"&STR$(CA)&"."&Q$,OUTPUT,SEQUENTIAL,INTERNAL,VARI
ABLE 254 :: PRINT #1:((J+1)-L):: FOR F=L TO J :: PRINT #1:C$(F):: NEXT F :: CLOS
E #1 :: RETURN
3691 REM
3711 PRINT
3721 INPUT "Nro. disco: 1,2,3 ?":CA :: IF (CA<1)+(CA>3)=0 THEN 3771
3741 PRINT :: CALL SOUND(500,220,1,659,1):: GOTO 3721
3771 PRINT : :"ENGANCHE TECLA ALPHA LOCK   Entre el nombre del titulo." :: INPUT
 "(no mas de 10 caracteres)   ":Q$ :: IF (LEN(Q$)<1)+(LEN(Q$)>10)=0 THEN 3831
3801 PRINT :: CALL SOUND(500,220,1,659,1):: GOTO 3771
3831 RETURN
3861 ON ERROR 3921 :: CALL ERR(DA,EA)
3871 CALL SOUND(500,110,1,220,1,659,1):: IF DA=39 OR DA=40 THEN 3891 ELSE 3901
3891 C$(I)="" :: C$(I-1)="" :: C$(I-2)="" :: C$(I-3)="" :: C$(I-4)="" :: I=I-4 :
: PRINT : :G$ :: GOTO 3941
3901 IF DA>82 AND DA<131 THEN PRINT : :"LO LAMENTO,ERROR I/O " :: GOTO 3941
3911 PRINT : :"<<< ERROR @#%! >>>" :: GOTO 3941
3921 CALL ERR(DA,EA,FA,GA):: PRINT "ERROR";DA;"EN LINEA";GA
3941 ON ERROR 3961
3951 CLOSE #1 :: GOTO 3971
3961 CALL ERR(DA,EA)
3971 RETURN 451
3981 CALL CLEAR
```

MUSICA EN LA TI 99

TI 99/4A




```
2 CALL CLEAR :: FOR A=1 TO 10 :: CALL COLOR(A,7,2):: NEXT A :: CALL SCREEN(2)::
CALL COLOR(9,7,2):: CALL CHAR(33,"00000000F0F0F0F0"):: CALL CHAR(34,"F0F0F0F0")
3 CALL CHAR(35,"FFFFFFFFF0F0F0F0"):: CALL CHAR(36,"F0F0F0F0F0F0F0F0"):: CALL CHA
R(37,"FFFFFFFFFFFFFFFF"):: CALL CHAR(38,"FF00FF00FF00FF00")
4 CALL CHAR(40,"F000F000F000F000"):: CALL VCHAR(4,3,37,2):: CALL VCHAR(4,6,37,2)
:: CALL VCHAR(4,9,37,2):: CALL HCHAR(5,4,33):: CALL HCHAR(4,4,34)
5 CALL HCHAR(3,3,35):: CALL HCHAR(3,4,34):: CALL HCHAR(5,5,33):: CALL HCHAR(3,6,
35):: CALL HCHAR(3,7,36):: CALL HCHAR(4,7,34)
6 CALL HCHAR(5,8,33):: CALL HCHAR(3,9,36):: CALL HCHAR(5,11,33):: CALL HCHAR(2,9
,34):: CALL HCHAR(24,1,38,30):: CALL VCHAR(1,30,40,24)
7 A$="EMPRESA PARA INFORMATICA" :: B=8 :: C=3 :: GOSUB 10 :: A$="PROGRAMAS PARA
EL" :: B=12 :: C=3 :: GOSUB 10 :: A$="MICROCOMPUTADOR TI-99/4A" :: B=14 :: C=3 :
: GOSUB 10
8 A$="FABRICADO Y DISTRIBUIDO  EN" :: B=18 :: C=3 :: GOSUB 10 :: A$="LA REPUBLIC
A  ARGENTINA POR" :: B=20 :: C=3 :: GOSUB 10 :: A$="'SDT'INTELIGENCIA ARGENTINA"
 :: B=22 :: C=3
9 GOSUB 10 :: FOR D=1 TO 700 :: NEXT D :: GOTO 11
10 FOR D=1 TO LEN(A$):: B$=SEG$(A$,D,1):: E=ASC(B$):: CALL HCHAR(B,C-1+D,E):: NE
XT D :: RETURN
11 ON WARNING NEXT :: CALL CHARSET :: CALL CLEAR
120 CALL CLEAR :: F=77 :: CALL SCREEN(14):: FOR G=1 TO 14 :: CALL COLOR(G,16,1):
: NEXT G :: FOR H=96 TO 143 :: READ C$ :: CALL CHAR(H,C$):: NEXT H :: DISPLAY AT
(5,14):CHR$(96)
150 DISPLAY AT(6,13):CHR$(97)&CHR$(99)&CHR$(98):: DISPLAY AT(7,12):CHR$(97)&"
"&CHR$(98):: CALL HCHAR(7,15,99,3)
170 DISPLAY AT(8,11):CHR$(97)&"     "&CHR$(98):: CALL HCHAR(8,14,99,5):: DISPLAY
 AT(9,10):CHR$(97)&"       "&CHR$(98):: CALL HCHAR(9,13,99,7)
190 DISPLAY AT(10,9):CHR$(97)&"         "&CHR$(98):: CALL HCHAR(10,12,99,9):: FO
R I=11 TO 13 :: CALL HCHAR(I,11,99,11):: NEXT I
200 CALL HCHAR(14,11,99,11):: FOR H=100 TO 106 :: CALL HCHAR(14,H-87,H):: NEXT H
210 DISPLAY AT(15,9):CHR$(99)&CHR$(99)&CHR$(107)&CHR$(108)&CHR$(109)&CHR$(110)&C
HR$(111)&" "&CHR$(113)&CHR$(99)&CHR$(99)
220 CALL HCHAR(16,11,99,11):: FOR H=114 TO 121 :: CALL HCHAR(16,H-102,H):: NEXT
H :: CALL HCHAR(17,11,99,11):: FOR H=122 TO 129 :: CALL HCHAR(17,H-110,H):: NEXT
 H
240 DISPLAY AT(18,9):CHR$(99)&CHR$(122)&CHR$(130)&CHR$(112)&CHR$(131)&CHR$(132)&
CHR$(133)&CHR$(112)&CHR$(134)&CHR$(99)&CHR$(99)
250 CALL HCHAR(19,1,142,32):: FOR H=135 TO 141 :: CALL HCHAR(19,H-122,H):: NEXT
H
260 DISPLAY AT(21,1):"MUSICA EN LA TI-99/4A      ======================      OP
RIMA UNA TECLA........."
270 F=F+4 :: CALL SPRITE(#1,143,16,76,F,-37,0,#2,143,16,76,F,-24,0):: READ J ::
IF J=880 THEN F=77 :: RESTORE 1640 :: GOTO 270
280 CALL KEY(0,K,L):: IF L<>0 THEN 300
290 FOR M=1 TO 5 :: CALL SOUND(-900,J,M*5,J*2,M*5,J*7.5,30,-4,M*5):: NEXT M :: C
ALL DELSPRITE(ALL):: GOTO 270
300 CALL CLEAR :: CALL DELSPRITE(ALL):: DISPLAY AT(1,7):"PARA HACER MUSICA EN LA
     TI-99/4A,SOLO SIGA LAS INS- TRUCCIONES.LAS TECLAS SON  LOS NUMEROS (0-9)."
310 DISPLAY AT(8,1):"ESTAS TECLAS SE CORRESPONDENCON LAS DE UN INSTRUMENTO." ::
DISPLAY AT(12,1):"VEAMOSLO:":"  1= DO":"  2= RE":"  3= MI"
330 DISPLAY AT(16,1):"  4= FA":"  5= SOL":"  6= LA":"  7= SI":"  8= DO":"  9= RE
":"  0= MI" :: DISPLAY AT(24,1):"OPRIMA CUALQUIER TECLA"
350 CALL KEY(0,K,L):: IF L=0 THEN 350
360 CALL SOUND(100,880,0):: CALL SOUND(100,1320,0)
370 N=-249 :: O=2^(1/12):: CALL CLEAR :: DISPLAY AT(10,1):"OPRIMA:
            R) PARA VER MENU          1) ORGANO                 2) ARMON
ICA"
390 DISPLAY AT(14,1):"          3) CLAVICORDIO              4) BOOGIE
     F) FIN"
400 CALL KEY(0,K,L):: IF L=0 THEN 400
410 CALL SOUND(499,880,17):: CALL SOUND(499,1320,17):: IF K=70 THEN END
430 IF K=82 THEN 370
440 IF K<49 OR K>52 THEN 400
450 ON K-48 GOTO 460,670,870,1080
460 CALL CLEAR :: DISPLAY AT(1,10):"ORGANO" :: DISPLAY AT(24,1):"OPRIMA 'R' PARA
 VER EL MENU"
470 DISPLAY AT(4,1):"  1 2 3 4 5 6 7 8 9 0":"   ^ ^ ^ ^ ^ ^ ^ ^ ^ ^":"  A B C
D E F G A B C" :: DISPLAY AT(23,1):"OPRIMA 'N' PARA OTRO TEMA." :: P=0 :: GOSUB
1330
490 CALL KEY(0,K,L):: IF L=0 THEN 490
500 IF K<>78 THEN 530
510 CALL SOUND(200,880,0):: CALL SOUND(100,1320,0):: P=P+1 :: IF P=10 THEN P=0 :
: GOSUB 1330 :: GOTO 490
520 ON P GOSUB 1340,1350,1370,1380,1400,1420,1440,1460,1480,32767
530 IF K=82 THEN 370
540 IF K<47 OR K>57 THEN 490
550 ON K-47 GOTO 660,560,570,580,590,610,620,630,640,650
560 CALL SOUND(N,220,0,440,9):: GOTO 490
570 CALL SOUND(N,247,0,494,9):: GOTO 490
580 CALL SOUND(N,262,0,524,9):: GOTO 490
590 CALL SOUND(N,294,0,588,9):: GOTO 490
600 FOR Q=1 TO 1999 :: NEXT Q :: CALL SCREEN(14)
610 CALL SOUND(N,330,0,660,9):: GOTO 490
620 CALL SOUND(N,349,0,698,9):: GOTO 490
630 CALL SOUND(N,392,0,784,9):: GOTO 490
640 CALL SOUND(N,440,0,880,9):: GOTO 490
650 CALL SOUND(N,494,0,988,6):: GOTO 490
660 CALL SOUND(N,523,0,1046,6):: GOTO 490
670 CALL CLEAR :: DISPLAY AT(1,10):"ARMONICA" :: DISPLAY AT(24,1):"OPRIMA 'R' PA
RA VER EL MENU"
680 DISPLAY AT(4,1):"  1 2 3 4 5 6 7 8 9 0":"   ^ ^ ^ ^ ^ ^ ^ ^ ^ ^":"  A B C
D E F G A B C" :: DISPLAY AT(23,1):"OPRIMA 'N' PARA OTRO TEMA." :: P=0 :: GOSUB
1330
700 CALL KEY(0,K,L):: IF L=0 THEN 700
710 IF K=82 THEN 370
720 IF K<>78 THEN 750
730 CALL SOUND(200,880,0):: CALL SOUND(100,1320,0):: P=P+1 :: IF P=10 THEN P=0 :
: GOSUB 1330 :: GOTO 700
```

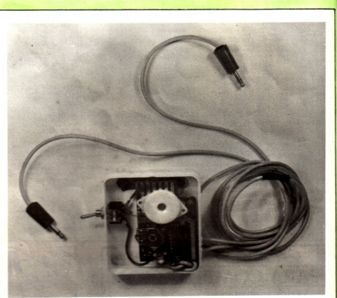


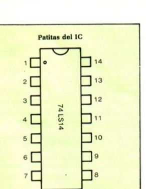
```
740 ON P GOSUB 1340,1350,1370,1380,1400,1420,1440,1460,1480,32767
750 IF K<47 OR K>57 THEN 700
760 ON K-47 GOTO 860,770,780,790,800,810,820,830,840,850
770 CALL SOUND(N,220,0,262,9):: GOTO 700
780 CALL SOUND(N,247,0,294,9):: GOTO 700
790 CALL SOUND(N,262,0,330,0):: GOTO 700
800 CALL SOUND(N,294,0,349,9):: GOTO 700
810 CALL SOUND(N,330,0,392,9):: GOTO 700
820 CALL SOUND(N,349,0,440,9):: GOTO 700
830 CALL SOUND(N,392,0,494,9):: GOTO 700
840 CALL SOUND(N,440,0,523,9):: GOTO 700
850 CALL SOUND(N,494,0,587,9):: GOTO 700
860 CALL SOUND(N,523,0,659,9):: GOTO 700
870 CALL CLEAR :: DISPLAY AT(1,9):"CLAVICORDIO" :: DISPLAY AT(24,1):"OPRIMA 'R'
PARA VER EL MENU"
880 DISPLAY AT(4,1):"  1 2 3 4 5 6 7 8 9 0":"  ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^":"  A B C
D E F G A B C" :: DISPLAY AT(23,1):"OPRIMQ 'N' PARA OTRO TEMA." :: P=0 :: GOSUB
1330
900 CALL KEY(0,K,L):: IF L=0 THEN 900
910 IF K<>78 THEN 940
920 CALL SOUND(200,880,0):: CALL SOUND(100,1320,0):: P=P+1 :: IF P=10 THEN P=0 :
: GOSUB 1330 :: GOTO 900
930 ON P GOSUB 1340,1350,1370,1380,1400,1420,1440,1460,1480,32767
940 IF K=82 THEN 370
950 IF K<47 OR K>57 THEN 900
960 ON K-47 GOTO 1060,970,980,990,1000,1010,1020,1030,1040,1050
970 R=220 :: K=440 :: GOTO 1070
980 R=247 :: K=494 :: GOTO 1070
990 R=262 :: K=524 :: GOTO 1070
1000 R=294 :: K=588 :: GOTO 1070
1010 R=330 :: K=660 :: GOTO 1070
1020 R=349 :: K=698 :: GOTO 1070
1030 R=392 :: K=784 :: GOTO 1070
1040 R=440 :: K=880 :: GOTO 1070
1050 R=494 :: K=988 :: GOTO 1070
1060 R=523 :: K=1046 :: GOTO 1070
1070 FOR F=0 TO 30 STEP 7 :: CALL SOUND(N,R,F,K,F):: NEXT F :: GOTO 900
1080 CALL CLEAR :: DISPLAY AT(1,10):"BOOGIE" :: DISPLAY AT(24,1):"OPRIMA 'R' PAR
A VER EL MENU"
1090 DISPLAY AT(5,1):"  1 2 3 4 5 6 7 8 9 0":"  ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^":"  A B C
 D E F G A B C"
1100 DISPLAY AT(12,1):"BOOGIE BOOGIE...": :"ES:    3   3   6   3             7
   6   3   0": :"OPRIMALAS DE A UNA......"
1110 CALL KEY(0,K,L):: IF L=0 THEN 1110
1120 IF K=82 THEN 370
1130 IF K<47 OR K>57 THEN 1110
1140 ON K-47 GOTO 1240,1150,1160,1170,1180,1190,1200,1210,1220,1230
1150 R=220 :: GOTO 1250
1160 R=247 :: GOTO 1250
1170 R=262 :: GOTO 1250
1180 R=294 :: GOTO 1250
1190 R=330 :: GOTO 1250
1200 R=349 :: GOTO 1250
1210 R=392 :: GOTO 1250
1220 R=440 :: GOTO 1250
1230 R=494 :: GOTO 1250
1240 R=523 :: GOTO 1250
1250 CALL SOUND(-N/2,R,0,R,0):: CALL SOUND(-N/2,R*2*D^0,0,R*D^0,0):: CALL SOUND(
N,R*D^4,0,.5*R*D^4,0):: CALL SOUND(N,2*R*D^4,0,R*D^4,0)
1260 CALL SOUND(N,R*D^7,0,.5*R*D^7,0):: CALL SOUND(N,2*R*D^7,0,R*D^7,0)
1270 CALL SOUND(N,R*D^9,0,.5*R*D^9,0):: CALL SOUND(N,2*R*D^9,0,R*D^9,0)
1280 CALL SOUND(N,R*D^10,0,.5*R*D^10,0):: CALL SOUND(N,2*R*D^10,0,R*D^10,0)
1290 CALL SOUND(N,R*D^9,0,.5*R*D^9,0):: CALL SOUND(N,2*R*D^9,0,R*D^9,0)
1300 CALL SOUND(N,R*D^7,0,.5*R*D^7,0):: CALL SOUND(N,2*R*D^7,0,R*D^7,0)
1310 CALL SOUND(N,R*D^4,0,.5*R*D^4,0):: CALL SOUND(N,2*R*D^4,0,R*D^4,0):: GOTO 1
110
1320 ON P GOSUB 1340,1350,1370,1380,1400,1420,1440,1460,1480,32767
1330 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"MARY TIENE UN CORDERITO..": :"E
S:    543 4555 444 577              543 4555 544 543!" :: RETURN
1340 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"BOGA BOGA EL BOTERO.....": :"ES
:     33345 54567                   0077  5533  765432" :: RETURN
1350 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"TITILA,TITILA, ESTRELLITA..": :
"ES:  3377 887                      6655 443"
1360 DISPLAY AT(16,1):"     7766 554                      7766 554
    3377 887                      6655 443" :: RETURN
1370 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"PUENTE DE LONDRES": :"ES:  7876
 567 456 567              7876 567 4753" :: RETURN
1380 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"CORRIENDO EN EL CAMPO.": :"ES:
 7757 875 54 54              7757 875 4565 43 3"
1390 DISPLAY AT(16,1):"     33570 88087                   7757 875 4565 43 3" :: R
ETURN
1400 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"ESTE VIEJO HOMBRE.": :"ES: 757
757 8765 456 5673 333 34567 744 65" :: RETURN
1420 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"VIEJO MACDONALD...": :"ES:    7
77 4554 99887              4 777 4554 99887"
1430 DISPLAY AT(16,1):"     4 777 777 7777 777            4 777 4554 99887" :: RET
URN
1440 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"CAMPANAS DE NAVIDAD.": :"ES:  5
55 555 57345               666 6655 5544 54 7"
1450 DISPLAY AT(16,1):"     555 555 57345                 666 6655 577 6 4 3" :: R
ETURN
1460 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"TRES CIEGUITOS.....": :"ES:  54
3 543 765 765              7009 89077"
1470 DISPLAY AT(16,1):"     7009 89077                    7009 89077
     6 5 4 3" :: RETURN
1480 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"HACE MUCHO TIEMPO.": :"ES:  3 3
45 567 875                 7 654 6 543"
1490 DISPLAY AT(16,1):"     3 345 567 875                 7 654 54 3" :: RETURN
1500 DATA 00000000444AAAEE,040A0A0E4EAEAEEE,40A0A0E0E4EAEAEE,EEEEEEEEEEEEEEEE,EE
EEEEEEEEEEEFEE,EEEEEFEBF0C08,EFF,FF,FE0F,EEEEEE1E0E0201
1520 DATA EEEEEEEEEEEEEE6E,ECEBF0F0E0E0E0C,0000000000000001,00010F1F356BD6D6,00F
FEFC78301,0000E0F05BACD6D7,FFFFFFFFFFFFFFFF,2E1E0E0E06060602
1540 DATA EEEEEFEFEFEFEFEF,C080808183838307,010303FFFF2900FF,AB05FFFFFF5200FF,01
83EFFFFFF9500FF,AB57FFFFFF2900FF
1560 DATA 008080FFFF5200FF,02010101B1C1C1E,EFEFEFEFEFEFEFEF,0F1F35303F203F3F,FFF
F2A00FF00FFFF,FFFF5200FF00FFFF,FFFFA500FF00FFFF
1570 DATA FFFF2A00FF00FFFF
1590 DATA FFFF5200FF00FFFF,F0F8AC0CFC04FCFC,3F3F3F3F3F3F3F3F,FFFFFAF5EAD5AA55,FF
55AA55AA55AA55AA,FFFFBF57AB55AAFD,FCFCFCFCFCFCFCFC
1610 DATA BFBFBFBFBFBFBFBF,FEFFFFFEFEFCFCF8,AA55FFAAAAAA2222,AB55FFABABAB202,F29
1CF4B243C,FF7FFF7F7F3F3F1F
1630 DATA FDFDFDFDFDFDFDFD,FF00FF00FF00FF,40E0E0E04
1640 DATA 110,123,131,147,165,175,196,220,247,262,294,330,349,392,440,494,523,58
7,659,698,784,880,999
```

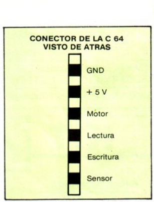
INTERFACE DE GRABADOR

Días pasados me compré una C64, pero lamentablemente no me alcanzó para el datassette, por lo que me puse a ver cómo podía conectar la máquina a un grabador común, para lo que tuve que desarmarle el datassette a mi amigo Sergio ante la desazón de su pequeño vástago, que temía no poder disfrutar más de guerras intergalácticas ni conducir fórmulas uno.

Como fruto de varios experimentos les ofrezco este circuito (que también se puede adquirir como kit). Reemplaza sin problemas al costoso datasette que, a pesar de ser de baja calidad, cuesta lo que un super walkman con am-fm, reloj y antena eléctrica.
El circuito usa el mismo integrado TTL que el datassete, un 74LS14 que es un schmitt-trigger séxtuple que transforma las ondas de sonido haciéndolas bien cuadraditas, para que la máquina interprete los unos y ceros. En el circuito de lectura se puso una llave, porque algunos grabadores invierten en la salida la polaridad de la señal; por lo que con la llave seleccionamos si la leemos tal cual, o la invertimos. En la primera etapa se pusieron dos resistencias para sumarle una tensión contínua a la señal y ubicarla en la zona de disparo de 74LS14. También se dispuso en forma opcional una bocina piezoeléctrica de alta impedancia que sirve para escuchar lo que lee la máquina (la bocina se encuentra en varias casas de electrónica del centro).
El circuito de grabación tiene un preset para ajustar el nivel de grabación por las dudas sea muy bajo o alto para nuestro grabador. (En la primera prueba ponerlo al medio). Como elemento opcional se agregó un relé de 6 voltios para controlar el remoto. No se puede en general usar un transistor aquí porque el remoto en muchos grabadores corta entre positivo y el borne del motor y si tuviéramos masa común produciríamos un cortocircuito en la fuente del grabador. El montaje puede hacerse en una plaquetita de experimentación y meterla en una cajita de plástico.





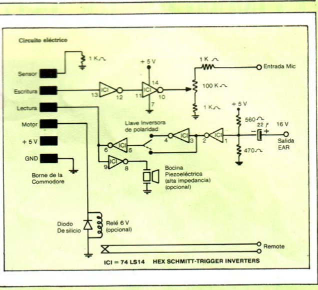


AJUSTE

Con la máquina apagada enchufamos el artefacto (luego de haber supervisado las conexiones), escri-

PARA COMMODORE

bimos un pequeño programita y tratamos de salvarlo (save "nombre"). Si el grabador permite monitorear por la salida de audífono, se escuchará la grabación en la bocina. Si no se escucha debe estar muy bajo el volumen. Luego rebobinamos y tratamos de leerlo; por la bocinita se escuchará el piii...crunch-crunch. Si no escuchamos nada, levantemos el volumen del grabador. Si tampoco se escucha levantar el volumen de grabación con el preset y volver a grabar.
Una vez que ya escuchamos el ruido, ponemos la cinta al principio y le damos load "". Si no reconoce el programa, volver de nuevo la cinta y probar con la llave para el otro lado. Cada grabador requiere una posición particular de la llave de lectura.



NOTAS

Si cargamos programas con turbo y no le pusimos el relé, una vez que le dimos el LOAD apretemos continuamente una tecla hasta que lo encuentra, porque si lo encuentra y no lo aceptamos enseguida se pasa de largo.
En un próximo artículo veremos la posibilidad de conectar impresoras comunes a la interfase serie de la C64 y todo otro pedido o idea que me acerquen.
Hasta la próxima.

Marcelo D. Martínez





S U M A R I O

AVANZADA TECNOLOGIA

PROGRAMAS INEDITOS



INTERFACE PARA COMMODORE

CARTA DEL DIRECTOR

Aunque parezca obvio decirlo en esta revista, queremos remarcar que las microcomputadoras "hogareñas" sirven no sólo para jugar sino que permiten una gran cantidad de aplicaciones prácticas. Un experto en estas cuestiones recordaba que un equipo de pocos K de memoria era utilizado, no hace mucho tiempo, por el Banco Central de la República Argentina para ordenar las finanzas del país. O sea que a una máquina del tipo de las que nos ocupamos en K 64 se le puede sacar "jugo" y aprovecharla en campos tan diversos como la educación, la contabilidad, la abogacía o la medicina. Ya hemos dado algunos ejemplos, y en esta edición continuamos ofreciendo programas utilitarios. Uno de ellos fue preparado por un juez para evaluar a alumnos de la carrera de Derecho. Otro ayuda a calcular los "sueldos temporarios".
Y en nuestro afán de darle soluciones a los usuarios de computadoras, publicamos en exclusividad dos desarrollos: uno soluciona el problema de software de las TS 2068, mientras que el otro sirve para conectar un grabador común a la C 64.
Y muchas otras notas más que nos colocan en el camino que nos hemos propuesto para presentar el mejor material inédito en una revista de nivel internacional.

CRISTIAN PUSSO

CONVERTIMOS LA TS 2068

COMO SACARLE JUGO A LA CZ Y TK

OTRAS NOTAS

K64
COMPUTACION PARA TODOS

AÑO 1 Nº 5 AGOSTO DE 1985

Director General
Ernesto del Castillo
Director Editorial
Cristian Pusso
Director Periodístico
Fernando Flores
Director Financiero
Javier Campos Malbrán
Secretaria
Moni Ocampo

Departamento de Publicidad:
Jefe: Dolores Urien
Promotora: Mónica Garibaldi
Departamento de Avisos
Oscar Devoto
Diagramación y Armado
Fernando Amengual y
Carlos Boccardo
Fotografía
Juan José Péres
Esteban Figueredo

K-64 es una Revista mensual editada por PROEDI Editorial S.A. (e./f.), Cerrito 1320, 1º Piso, Buenos Aires, Te.: 42-9681/9. Registro Nacional de la Propiedad Intelectual: 313.837 M. registrada

Precio de este ejemplar: un austral con cincuenta centavos. Precio de la suscripción: semestral: 8 australes.
Distribuidor en Capital: Infinito. Venezuela 1417 Capital Federal. Tel.: 37-6664.
Impresión: Calcotam. Fotocromo tapa: Columbia.
Fotocomposición: Van Waveren.
Los ejemplares atrasados se venderán al precio del último número en circulación.

CORREO ARGENTINO CASA CENTRAL
FRANQUEO PAGO CONCESION Nº 2538
TARIFA REDUCIDA CONCESION Nº 836

EL SISTEMA OPERATIVO DEL COMMODORE 1541



El Commodore 64, como comentáramos en su presentación (K 64 Nº 2, pag. 28), es un ordenador de aceptables prestaciones en el nivel de gestión administrativa. En este artículo, efectuaremos una revisión de las utilidades que se pueden obtener empleando unidades de mini discos Commodore 1541 y, en particular del funcionamiento de su programa controlador o "Sistema Operativo".

Luego de meses de utilizar el lentísimo grabador a cassettes ("Quelonio", como algunos lo han bautizado), probablemente habrá empezado a considerar muy seriamente la posibilidad de adquirir una unidad de discos. Es un hecho que el tiempo de carga de programas desde el datassette resulta francamente exasperante y, al mismo tiempo, su utilización nos impide sacar provecho de avanzados sistemas de bases de datos disponibles en el mercado (Superbase 64, The Manager, etc.) por la reducida capacidad de manejo de archivos que tiene este periférico.

Ahora bien, efectuada la adquisición de su nuevo periférico y una buena provisión de disquettes, Ud. se encontrará con un manual del usuario notablemente escueto y caracterizado por una buena tanda de errores (de hecho, la mayor parte de los manuales de periféricos CBM son muy breves). Dejando de lado los capítulos sobre instalación y encendido, consideramos que la descripción de los co-

mandos del Sistema Operativo de la Disquettera (D.O.S. o Disk Operating System) es muy reducida y la carencia de ejemplos extensos para cada comando desanima al usuario a emplearlos.

LOS COMANDOS DEL D.O.S.:

Veamos a continuación los comandos disponibles en el DOS Versión 2 de las disquetteras Commodore 1541:

OPEN (ABRIR):

Excepto al efectuar la carga, grabación o verificación de un programa; toda operación hacia la disquetera debe iniciarse con esta orden, que dice al D.O.S. que abra un archivo y se prepare a leer o grabar datos, a recibir información sobre el estado del periférico u órdenes especiales. Su sintaxis es:
OPEN NA,ND,NC [,"TEXTO"]
Donde NA es el número de archivo dentro de la computadora (también podemos llamarle "vía"); ND es el número de dispositivo, es decir el número que identifica a la unidad de discos; NC es el número del canal dentro de la disquettera (también podemos llamarle "dirección secundaria") y "TEXTO" es una sarta de caracteres que se emplea o no según el tipo de apertura de archivos que se efectúe.
NA puede valer cualquier número de 1 a 255; ND puede valer de 8 a 11 si bien, por defecto, es 8. Si tuviera necesidad de cambiar dicho valor vea el Capítulo 9 de su manual del usuario para modificarlo apropiadamente. Recomendamos efectuar la modificación por el método de software y no por hardware, ya que la apertura y manipulación del contenido de la disquettera, sin los conocimientos necesarios, pueden traer resultados terribles.
NC, como ya dijimos, es el número de canal por el cual pasará la información dentro de la disquettera. En la 1541 existen 16 de estos canales; pero tres de ellos tienen funciones especiales: los Nº 0 y 1 están reservados para operaciones LOAD, SAVE y VERIFY; y el Nº 15 es conocido como "Canal de Comando" ya que él es el que sirve para dar órdenes específicas al drive y para que éste nos informe el resultado de esas órdenes.
La última parte de la sentencia OPEN, el "TEXTO", varía respecto de la apertura que se realice; veamos a continuación una tabla de comandos OPEN:

SENTENCIA	COMENTARIOS
OPEN 15,8,15	Abre el canal de comando del dispositivo Nº 8. A continuación, se podrá dar una orden. al D.O.S. por medio de una sentencia PRINT# que veremos más adelante; o bien leer el estado del dispositivo (lectura de condiciones de error).
OPEN 2,8,2"SEC,S,R"	Abre el archivo Nº 2 hacia el canal 2 del drive número 8. Allí opera sobre el archivo llamado "SEC"; que es de tipo S (secuencial) y que se va a leer (la R indica lectura).
OPEN 3,8,5"K64,S,W"	Abre el archivo número tres hacia el canal número 5 del dispositivo 8. Allí escribirá sobre el archivo secuencial K64 (la última W denota escritura).
OPEN 2,8,2"$"	Abre el archivo número 2 al canal número 2 del dispositivo 8. Allí opera sobre el directorio del disco, que se denomina "$". Mientras OPEN está en vigencia es muy recomendable NO efectuar una orden PRINT#2 ya que ésta destruirá parte del directorio.
OPEN 8,8,8,"#"	Abre el archivo número 8 hacia el canal del mismo número en el drive 8. En este caso, el texto "#" indica que se abre un archivo de tipo "Random" (de acceso aleatorio), que se caracteriza por permitir la lectura de un bloque completo del disco (256 bytes).
A$="REL,L,"+CHR$(12)	Asigna a la variable A$ el nombre de un archivo relativo más la sarta ",L," y el CHR$(12) que indica la longitud de cada registro de este archivo.
OPEN 7,8,3,A$	Abre el archivo 7 al canal 3 del dispositivo 8. El texto A$ le indica que se está creando un archivo relativo, en el cual cada registro tendrá una longitud de 12 bytes.
TX$=CDOS$+":"+ARCH$	Asigna a la variable TX$ un comando (CDOS$); los dos puntos y el nombre de un archivo (ARCH$).
OPEN 15,8,15,TX$	Abre el archivo 15 al canal de comando de la disquettera 8 y le ordena ejecute el comando CDO$ sobre el archivo ARCH$.

En la sintaxis que describimos más arriba; y como demostramos en estos ejemplos, todos los componentes de una sentencia OPEN (NA, ND, NC y "TEXTO") pueden ser variables. Al usar estos parámetros con variables se debe tener la precaución de asegurar la asignación de valores correctos a las mismas. De lo contrario, pueden generarse condiciones de error tales como la de "DEVICE NOT PRESENT ERROR" (error de dispositivo ausente). Por otro lado, es muy recomendable utilizar el mismo número de archivo y de canal; para de este modo tener clara la relación entre ambos. Así, por ejemplo, el archivo 15 siempre estará relacionado al canal de error de la disquettera.

CLOSE (CERRAR):

Concluídas las operaciones con una disquettera (excepto LOAD, SAVE y VERIFY) es imperativa la ejecución del comando CLOSE que cerrará el/los archivos sobre los cuales se estuvo operando. Su sintaxis es muy simple:

CLOSE NA

Donde NA es el número del archivo que se desea cerrar. NA puede ser una constante o una variable; y aquí también valen las consideraciones hechas sobre la sentencia OPEN. Cerrar un archivo que nunca se abrió dará por resultado un mensaje "FILE NOT OPEN ERROR" (error de archivo no abierto).

SENTENCIAS DE ENTRADA/SALIDA:

Cuando se trabaja con archivos en

discos, la entrada/salida de datos desde las disquetteras es idéntica a la proveniente del teclado y se efectúa por medio de órdenes INPUT#NA, GET#NA y PRINT#NA, siendo NA el número del archivo desde/hacia el cual se efectúan estas entradas/salidas. Obviamente, el archivo NA deberá haber sido abierto antes de efectuar la primera de cualquiera de estas operaciones.

La sentencia INPUT# lee bytes hasta encontrar un CHR$(13) (retorno de carro) y asigna los bytes leídos anteriormente a la variable que se le indique. La sentencia GET#, en cambio, lee un byte y lo asigna a la variable especificada sin discriminar sus contenidos; salvo en el caso en que se intente leer un byte que define un caracter alfabético y cargarlo en una variable numérica. La orden PRINT# imprimirá en el archivo especificado el contenido de la variable seguido de un CHR$(13) o no, según como se cierre la sentencia.

Cuando se lee un archivo del cual se desconoce el formato de sus datos, se recomienda utilizar la sentencia GET# y una variable alfanumérica y analizar uno por uno los bytes leídos desde el archivo para determinar el contenido real del archivo y descartar apropiadamente los separadores de registros. A continuación, veamos algunos ejemplos de instrucciones de entrada/salida:

NEW (NUEVO):

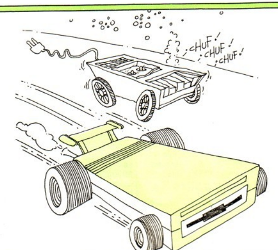
Este comando se utiliza para formatear o reformatear un disco. Según la forma en que se emplee, borrará todo el contenido del disquette ya que "dibujará" sobre la capa magnética del mismo los bloques en los que posteriormente grabará sus programas y datos. Para ordenar este comando utilizamos las sentencias OPEN y CLOSE de la siguiente forma:
OPEN 15,8,15,"N:"+NDS+","+ID $:CLOSE15
Donde ND$, el nombre del disquette, es una sarta de hasta 16 caracteres de longitud e ID$, que representa el identificador del mismo, es otra de 2 caracteres de largo. Dicho comando formateará el disquette identificándolo con el nombre contenido en ND$ y el identificador descripto en ID$. Si el disquette ya estaba formateado toda la información que éste contenía se perderá definitivamente.
En ciertas oportunidades, Ud. deseará re-formatear un disco que ya no tiene mayor utilidad pero que estaba formateado. Para ganar tiempo en este proceso, el comando de formateo a ordenar será:
OPEN 15,8,15,"N:"+ND$:CLOSE15
Una vez más, ND$ representa el nombre del disco y puede tener hasta 16 caracteres. El formateo de un disco de acuerdo al último procedimiento aquí descrito no destruye la información contenida en él; con la única excepción del directorio ($) que sí es efectivamente borrado. Por esta razón, se puede recuperar la información de ciertos archivos antiguos si el disco reformateado no se ha usado demasiado.
Una recomendación personal que hacemos a los lectores es la de utilizar nombres de discos normalizados, es decir, denominados de una forma estandarizada. Por ejemplo, todos nuestros discos siguen la convención de tener como nombre un número de cuatro dígitos rellenado con ceros a la izquierda y como identificador el número de disco representado en dos dígitos hexadecimales. De esta manera, el disco más antiguo lleva el nombre 0001 y el identificador 01. Algunos dirán, y qué pasa cuando lleguen al disco número 255 (FF en hexadecimal)?. Lo más probable es que, cuando lleguemos a dicha unidad, el número 0001 ya estará fuera de servicio por su excesivo uso. De este modo el ciclo se repetirá.
Otra idea sería utilizar la fecha de formateo inicial como nombre del disco. Así, un disquette formateado el 30 de junio de 1985 podría llevar el nombre "85-06-30" o alguna combinación similar. En este caso, aparte de un nombre siempre distinto para cada disco, tendremos información sobre cuanto tiempo ha estado el mismo en servicio; información que nos permitirá determinar el momento propicio para su reemplazo.

COPY (COPIAR):

Este es un comando que, dentro del mismo disco, tiene la capacidad de copiar un archivo e, inclusive, de encadenar hasta cuatro archivos en uno. En el primer caso, su utilidad es reducida ya que no tiene mayor sentido contar con dos versiones del mismo archivo en el mismo disco. Por el contrario, la segunda opción es muy beneficiosa ya que nos permite encadenar diversos archivos y generar un archivo principal.
Supongamos que tenemos cuatro archivos secuenciales que contienen datos referidos a los clientes de cuatro sucursales distintas de una misma empresa de computación; y que la gerencia de ésta desea un informe completo sobre toda su clientela. Una forma de preparar dicho informe sería tratar por separado cada uno de los archivos de las sucursales. Otra solución más económica desde el punto de vista del programador sería encadenar esos cuatro archivos en uno

Lo que sigue es de Ejemplos de ENTRADA/SALIDA:

SENTENCIA	COMENTARIOS
GET#5,A$	Toma un byte del archivo Nº 4 y lo asigna a la variable sarta A$.
GET#3,A	Toma un byte del archivo Nº 3 y su valor real lo asigna a la variable A.
GET#7,A$,B$,A,D,D%	Lee cinco bytes y los asigna según su tipo a las variables especificadas.
INPUT#2,A$	Lee una serie de bytes y los concatena en A$ hasta hallar un CHR$(13). Si esto no sucede hasta el byte número 80, el programa BASIC se interrumpe con un mensaje "STRING TOO LONG" (sarta de caracteres demasiado larga).
INPUT#2,A	Lee una serie de bytes hasta hallar un CHR$(13) o un valor incorrecto para una variable real.
PRINT#3,F;F$	Imprimirá en el archivo 3 la variable A incluyendo un espacio a la izquierda para su signo seguida de un CHR$(13) e, inmediatamente, el contenido de la variable alfanumérica F$.



provisorio y luego procesar este último.
De esta manera se reduciría el potencial dado que cualquiera de los cuatro archivos separados podría sufrir. Al mismo tiempo, el programador tendría menos que codificar ya que al principio del programa uniría los archivos y luego se dedicaría a procesar el principal.
Para copiar un solo archivo en otro la sentencia es:
OPEN 15,8,15,"C:"+NA$+"="+" VA$"
Donde NA$ contiene el nombre de la copia y VA$ el nombre del original.
En el caso de encadenamiento múltiple, el comando será:
OPEN 15,8,15,"C:"+NA$+"="+ VA$(1)+","+VA$(2)+","+VA$(3)+ ","+VA$(4):CLOSE15
Donde NA$ contiene el nombre del archivo de destino y el arreglo VA$ los archivos originales a encadenar.

RENAME (RENOMBRAR):

Este comando permite cambiar el nombre de un archivo existente en el directorio del disco. Para efectuar este cambio sólo se debe ingresar:
OPEN 15,8,15,"R :"+NN$+"="+ VN$:CLOSE15
Donde NN$ contiene el nuevo nombre que le queremos asignar al archivo y VN$ su nombre actual.

SCRATCH ("RASCAR"):

Este comando, al que nosotros llamamos "KAPUTT", tiene como función la "destrucción" de un archivo. En realidad, la ejecución de este comando no destruye el archivo ni su referencia en el directorio. Lo que si hace es marcarlo como "nulo" o "borrado" (deleted) en su referencia del directorio y hacer que los bloques que ocupa queden disponibles para otro archivo. Esto implica que un archivo que ha sido borrado con este comando por error, se puede recuperar; por supuesto si el disco no fue usado demasiado luego del borrado. Su sintaxis es:
OPEN 15,8,15,"S:"+NA$:CLOSE15
Donde la variable NA$ contiene el nombre del archivo al que se aplicará este comando.
De acuerdo a experiencias de otros usuarios, nos hemos enterado que en ciertas ocasiones el comando SAVE & REPLACE (vea su manual para la sintaxis apropiada) no funciona correctamente; e inclusive ha llegado a destruir algunos archivos. Para evitar el potencial efecto nocivo de dicho comando, nosotros utilizamos una breve rutina que cumple la función del SAVE & REPLACE con absoluta confiabilidad. En nuestros programas usamos esta rutina, ubicándola en las primeras líneas, para asegurarnos que la última versión del programa en desarrollo haya sido grabada al empezar su ejecución. De este modo, si durante el programa se produce una interrupción inesperada, de la cual no podemos recuperarnos, aún tendremos el programa en el disco.
Como alternativa, esta rutina se puede colocar en alguna parte del programa que habitualmente no se ejecuta; tal el caso de una subrutina que nunca se llama desde el programa o de varias líneas al final del mismo precedidas por una sentencia STOP. De esta manera, desde el modo directo se podrá operar esta rutina sólo las veces que se desee; con el consiguiente ahorro de tiempo que esto implica. Las tres posibilidades aquí comentadas se muestran en el listado Nº 2.

INITIALIZE (INICIALIZAR):

Este comando es a la disquettera 1541 lo que la SYS 64738 es a la Commodore 64; es decir, coloca a la máquina en la condición que ésta tiene al momento de ser encendida. En los drives se utiliza cuando ciertas condiciones de error particularmente extrañas nos impiden operarlo en forma normal. Para efectuar una inicialización, se debe entrar la orden:
OPEN 15,8,15,"I":CLOSE15

VALIDATE (VALIDAR):

Este último comando del D.O.S. sirve para "ordenar" un disco que, a raíz de numerosas operaciones de grabación y borrado, tiene una mala distribución de los bloques existentes en él. La única restricción a esta orden surje en aquellos discos que contienen archivos de tipo relativo, que son los más usados por los sistemas administradores de bases de datos. En caso de tener un disco de este tipo NUNCA efectúe una VALIDATE sobre el mismo. Para ejecutar este comando ingrese la orden:
OPEN 15,8,15,"V":CLOSE15

OTROS ITEMS DEL D.O.S.:

Como dijéramos al principio de esta nota, el canal de comando está preparado para entregarnos informes sobre sus condiciones de error. El manual de las disqueteras incluye una breve rutina BASIC para leer ese canal de error que nos informa el número código del error; su descripción en una breve leyenda; y la pista y sector en donde se detectó el mismo. Si se está empleando o desarrollando un programa de archivos es imprescindible contar con una subrutina que verifique este canal después de cada operación de entrada/salida. Al mismo tiempo, podemos utilizarla en modo directo para leer ese canal cada vez que vemos la luz roja de la disquettera titilar.
Por otra parte, existen en todo disco de demostración del 1541 dos

archivos de programa llamados "C-64WEDGE" y "DOS 5.1" que sirven de gran ayuda. El primero, como su nombre lo indica, es una cuña que carga el segundo. Este, una vez activado, brinda una serie de comandos directos que, en realidad, son una abreviatura de los aquí explicados (que pertenecen al D.O.S. Versión 2.0); más algunos habitualmente no disponibles. En el listado 4, exhibimos la correlación entre los comandos del D.O.S. 5.1 y el D.O.S. 2.0.
Para sacar mayor provecho del WEDGE y el D.O.S. 5.1 recomendamos copiar ambos al principio de todo disquette nuevo y, cada vez que se enciende la máquina, cargarlos y correrlos para tenerlos siempre activos. Si, una vez cargado el D.O.S. 5.1, se ejecuta una orden SYS 64738, no es necesario volver a cargar y correr el WEDGE, con sólo pulsar SYS 52224 se volverá a activar.
Adjunto a este artículo (listado 3) encontrarán Uds. el programa "Inspector de Directorios" (no suena a cargo de funcionario?) que sirve para obtener mayor información sobre los archivos contenidos en un disco. Una vez tipeado y grabado en disco corra el programa colocando un disco en el drive. Una vez hecho ésto pulse RETURN y el programa le hará conocer el nombre y tipo de cada archivo junto con la longitud del archivo en bloques (columna LAR); la primera pista (columna PI) y sector (columna SE) que éste ocupa y, si el archivo es relativo, la longitud de cada registro del mismo (columna LRG).

...Y ESTE ES EL BUS QUE SE VA:

Todos los comandos del D.O.S. que ejemplificamos en este artículo están en la así llamada "forma abreviada", es decir, se invocan usando sólo la inicial del comando. Si bien no tiene mayor aplicación, esos comandos pueden ser invocados utilizando su nombre completo. Por ejemplo, para validar un disco se puede ejecutar la sentencia OPEN 15,8,15,"V" cuyo efecto será idéntico a OPEN 15,8,15,"VALIDATE".
Por otra parte, los comandos del D.O.S. también se pueden ordenar por medio de sentencias PRINT# desde dentro de un programa y no en modo inmediato como los hemos ejemplificado. Para ello se deberá tener la precaución de abrir el canal de comando antes de efectuar cualquier comando por medio de un PRINT#. De lo contrario, el programa se detendrá con un mensaje de "FILE NOT OPEN ERROR" (el archivo no fue abierto).
De igual modo, no se deben usar las sentencias tal cual las ejemplificamos en esta nota una vez que el archivo fue abierto. Si se intenta abrir un archivo que ya ha sido abierto, el programa se interrumpirá con un "FILE OPEN ERROR" (el archivo ya está abierto).

CARLOS A. AY
y DANIEL H. MANDUCA

```
0 REM ************************************
1 REM OPCION 1: GRABA CADA VEZ QUE SE
2 REM           EJECUTA EL PROGRAMA.-
3 REM ************************************
4 :
5 NA$="PROGRAMA"
6 OPEN 15,8,15,"S:"+NA$:CLOSE15
7 SAVE NA$,8:VERIFY NA$,8
8 :
9 REM AQUI CONTINUA EL PROGRAMA
```

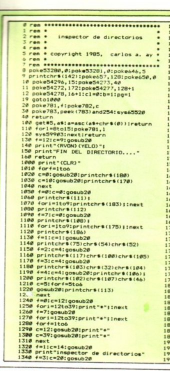
```
0 REM ************************************
1 REM OPCION 2: GRABA CADA VEZ QUE SE
2 REM           SE ORDENA GOSUB 100
3 REM           EN MODO DIRECTO O DESDE
4 REM           UN PUNTO DEL PROGRAMA.-
5 REM ************************************
70 :
80 :
96 :
100 NA$="PROGRAMA"
110 OPEN 15,8,15,"S:"+NA$:CLOSE15
120 SAVE NA$,8:VERIFY NA$,8
130 RETURN
131 :
140 REM AQUI SIGUE EL PROGRAMA
```

```
0 REM ************************************
1 REM OPCION 1: GRABA CADA VEZ QUE SE
2 REM           ORDENA GOTO 1000  EN
3 REM           MODO DIRECTO.-
4 REM ************************************
8999 :
9000 REM AQUI TERMINA EL PROGRAMA
9001 :
9999 STOP
10000 NA$="PROGRAMA"
10010 OPEN 15,8,15,"S:"+NA$:CLOSE15
10020 SAVE NA$,8:VERIFY NA$,8
```

LISTADO COMPARATIVO DE COMANDOS D.O.S. 5.1 VERSUS D.O.S. 2.0

COMANDO DOS 2.0	COMANDO DOS 5.1	OBSERVACIONES
LOAD"PROGRAMA",8	/PROGRAMA	
LOAD"PROGRAMA",8,1	%PROGRAMA	LENG.DE MAQUINA
LOAD"PROGRAMA",8:RUN	↑PROGRAMA	AUTO-RUN
SAVE"PROGRAMA",8	←PROGRAMA	
SAVE"@:PROGRAMA",8	←@:PROGRAMA	SAVE & REPLACE
(NO EXISTE)	@ 0 >	LEE CANAL ERROR
(NO EXISTE)	@$ 0 >$	LEE DIRECTORIO
PRINT#15,"C:COPIA=ORIGINAL"	@C:COPIA=ORIGINAL	COPIA
PRINT#15,"I"	@I	INICIALIZA
PRINT#15,"N:NOMBRE,ID"	@NOMBRE,ID	FORMATEA
PRINT#15,"Q"	@Q	ANULA DOS 5.1
PRINT#15,"R:NUEVO=VIEJO"	@R:NUEVO=VIEJO	RENOMBRA
PRINT#15,"S:ARCHIVO"	@S:ARCHIVO	BORRA ARCHIVO
PRINT#15,"U1"	@U1	REESTABLECE DOS
PRINT#15,"V"	@V	VALIDA

ESTE LISTADO ASUME QUE ANTES DE LAS SENTENCIAS QUE EMPIEZAN CON PRINT#15 SE ORDENA UN OPEN 15,8,15 Y QUE LUEGO DE ELLAS SE EFECTUA EL CORRESPONDIENTE CLOSE 15.



```
0 rem ********************************
1 rem *                              *
2 rem *   inspector de directorios   *
3 rem *                              *
4 rem *                              *
5 rem * copyright 1985,  carlos a. ay *
6 rem *                              *
7 rem ********************************
8 poke53280,0:poke53281,0:poke646,5
9 printchr$(142):poke657,128:poke650,0
10 poke54296,15:poke54273,40
11 poke54272,172:poke54277,128+1
12 poke54278,16+1:cl=0:b=1:pg=1
19 goto1000
20 poke781,f:poke782,c
30 poke783,peek(783)and254:sys65520
40 return
100 get#5,a$:a=asc(a$+chr$(0)):return
110 forl=8to15:poke781,l
120 sys59903:next:return
130 f=12:c=9:gosub20
140 print"(RVON)(YELO)";
150 print"FIN DEL DIRECTORIO...."
160 return
1000 print"(CLR)"
1010 forf=1to6
1020 c=0:gosub20:printchr$(180)
1030 c=10:gosub20:printchr$(170)
1040 next
1050 f=0:c=0:gosub20
1060 printchr$(111);
1070 fori=1to9:printchr$(183);:next
1080 printchr$(112)
1090 f=7:c=0:gosub20
1100 printchr$(108);
1110 fori=1to9:printchr$(175);:next
1120 printchr$(186)
1130 f=1:c=1:gosub20
1140 printchr$(75)chr$(54)chr$(52)
1150 f=2:c=4:gosub20
1160 printchr$(117)chr$(100)chr$(105)
1170 f=3:c=4:gosub20
1180 printchr$(103)chr$(32)chr$(104)
1190 f=4:c=4:gosub20:printchr$(106);
1200 printchr$(102)chr$(107)chr$(46)
1210 c=5:forf=5to6
1220 gosub20:printchr$(113)
12   next
1240 f=0:c=12:gosub20
1250 fori=12to39:print"*";:next
1260 f=7:gosub20
1270 fori=12to39:print"*";:next
1280 forf=1to6
1290 c=12:gosub20:print"*"
1300 c=39:gosub20:print"*"
1310 next
1320 f=1:c=14:gosub20
1330 print"inspector de directorios"
1340 f=3:c=20:gosub20
1350 print"escrito por:"
1360 f=5:c=18:gosub20
1370 print"carlos arturo ay"
1380 f=9:c=2:gosub20
1390 print"coloque en la unidad numero";
1400 print" 8 el disco"
1410 f=11:gosub20
1420 print"cuyo directorio desea";
1430 print" inspeccionar."
1440 f=13:gosub20
1450 print"cierre la puerta de la";
1460 print" unidad y pulse"
1470 f=15:gosub20
1480 print"<return> para continuar."
1490 ifcl=255thenb=-b
1500 cl=cl+b
1510 fori=55296to55576step40
1520 forj=itoi+10
1530 pokej,cl
1540 next:next
1550 poke54276,33:poke54276,32
1560 geta$:ifa$=""then1490
1570 print"(CLR)"chr$(14)
1580 fori=0to4:readtp$(i):next
1590 data  "nulo","sec.","prog."
1600 data "usuar.","relat."
1610 f=1:c=1:gosub20
1620 print"Nombre del disco:"
1630 f=2:gosub20
1640 print"Identificador:"
1650 f=3:gosub20
1660 print"D.O.S. Version:"
1670 f=4:gosub20
1680 print"Directorio"
1690 c=30:gosub20
1700 print"Pagina:"
1710 f=6:c=0:gosub20
1720 print"Nombre"
1730 c=18:gosub20
1740 print"Tipo"
1750 c=27:gosub20
1760 print"LAR PI SE LRG"
1770 f=7:c=0:gosub20
1780 fori=0to39:printchr$(163);:next
1790 f=16:gosub20
1800 fori=0to39:printchr$(163);:next
1810 f=18:c=1:gosub20
1820 print"Presione <RETURN> para ";
1830 print"ver la proxima"
1840 f=19:c=3:gosub20
1850 print"pagina. "
1860 f=20:c=1:gosub20
1870 print"Presione <ESPACIO> para ";
1880 print"revisar el di-"
1890 f=21:c=3:gosub20
1900 print"rectorio de otro disco. "
1910 p=18:s=0:pg=1
1920 open15,8,15
1930 open5,8,5,"#"
1940 print#15,"ua:"5;0;p;s
1950 gosub100:p=a:gosub100:s=a
1960 fori=2to143:gosub100:next
1970 nd$="":fori=0to17
1980 gosub100
1990 ifa<>160thennd$=nd$+a$
2000 next
2010 f=1:c=19:gosub20:print"(WHT)"nd$
2020 id$="":fori=1to3:gosub100
2030 id$=id$+a$:next
2040 f=2:c=16:gosub20:printid$
2050 so$="":fori=1to2:gosub100
2060 so$=so$+a$:next
2070 f=3:c=17:gosub20:printso$
2080 ifp=0thengosub130:goto2345
2090 print#15,"ua:"5;0;p;s
2095 f=4:c=37:gosub20:printpg
2100 gosub100:p=a:gosub100:s=a:f=7
2105 pg=pg+1
2110 fork=0to7
2120 gosub100:tp=aand7
2130 gosub100:pi=a
2140 gosub100:se=a
2150 na$="":fori=0to15:gosub100
2160 ifa<>160thenna$=na$+a$
2170 next
2180 fori=0to2:gosub100:next:lr=a
2190 fori=22to27:gosub100:next
2200 gosub100:ll=a:gosub100:lh=a
2210 l=256*lh+ll
2220 gosub100:gosub100
2230 ifna$=""then2310
2240 f=f+1:c=0:gosub20:printna$
2250 c=18:gosub20:printtp$(tp)
2260 c=26:gosub20:printl
2270 c=30:gosub20:printpi
2280 c=33:gosub20:printse
2290 iftp<>4then2310
2300 c=36:gosub20:printlr
2310 next
2320 getb$
2330 ifb$=chr$(13)thengosub110:goto2080
2340 ifb$<>chr$(32)then2320
2345 close5:close15
2350 forl=18to21:poke781,l
2360 sys59903:next
2370 print"(GRN)"
2380 f=19:c=1:gosub20
2390 print"Desea revisar otro disco";
2400 input"(C/RT)(C/RT)si(C/LF)(C/LF)(C/
LF)(C/LF)";re$
2410 ifleft$(re$,1)="s"thenrun
2420 f=21:c=1:gosub20
2430 print"Confirma salida";
2440 input"(C/RT)(C/RT)no(C/LF)(C/LF)(C/
LF)(C/LF)";re$
2450 ifleft$(re$,1)="n"thenrun
2460 end:rem sys64738
```

NUMERADOR AUTOMATICO DE LINEAS:

Comp.: Commodore 64.
Conf.: 64K con disquetera o grabador a cassettes.
Clas.: Utilitario.



Este utilitario, escrito totalmente en lenguaje de máquina, numera automáticamente la próxima línea de un programa BASIC cuando se pulsa la tecla RETURN. Por defecto, el programa genera números de línea a partir del 0 con incrementos de 10, aunque estos valores se pueden modificar. Además, el numerador incluye una función especial que junto con el número de línea entrega la leyenda DATA, lo que resulta particularmente útil para programas con abundantes sentencias de este tipo.

Para cargar este utilitario, hay que ingresar y grabar en disco o cassette el cargador BASIC adjunto. Debemos correr este booter, que se encargará de colocar el utilitario a partir de la posición 49152 de memoria, y luego pulsamos NEW. A continuación, ingresamos en modo directo la sentencia SYS 49152, con lo que se activará el numerador. El siguiente cuadro detalla el efecto de las teclas de función afectadas por el numerador automático:

- f1 enciende y apaga la función de numeración automática.
- f3 enciende y apaga la función de leyenda DATA automáticamente.
- f5 permite modificar el incremento a utilizar en la numeración automática.
- f7 permite modificar la línea desde donde comienza la numeración automática.

Por ejemplo, cargamos y ejecutamos el booter y tipeamos NEW y SYS 49152. Después pulsamos f1 y RETURN, de inmediato aparecerá la línea número 10. Pulsamos una sentencia cualquiera, pulsamos RETURN y veremos la línea 10. Pulsamos f3 y la línea siguiente comenzará como 20 DATA. Pulsamos f3 nuevamente y la línea 30 será normal (es decir, sin leyenda DATA).

Si deseamos comenzar en la línea 1000 (o cualquier otra), pulsamos f7, ingresamos el número de línea deseado y tecleamos RETURN. Pulsamos RETURN otra vez y dicha línea será la próxima a ingresar. Si queremos modificar el incremento entre número de línea, pulsamos f5 e ingresamos el mismo. A partir de ese momento, las líneas estarán numeradas de acuerdo al incremento pedido. Para desactivar la función de numeración automática, pulsamos f1 nuevamente y el editor volverá a su funcionamiento normal. Para desactivar totalmente el numerador pulsamos RUN/STOP y RESTORE al mismo tiempo. Si no apagamos la C-64, SYS 49152 volverá a activar el numerador.

```
0 REM ********************************
1 REM *                              *
2 REM *     NUMERADOR AUTOMATICO     *
3 REM *                              *
4 REM *  REVISTA K64, AGOSTO DE 1985  *
5 REM *                              *
6 REM ********************************
10 FORI=49152TO49537:READA:POKEI,A:NEXT
49152 DATA 173,36,3,201,188,240,19,141,
189,192
49162 DATA 173,37,3,141,190,192,169,188
,141,36
49172 DATA 3,169,192,141,37,3,173,20,3,
201
49182 DATA 55,240,21,141,102,192,173,21
,3,141
49192 DATA 103,192,120,169,55,141,20,3,
169,192
49202 DATA 141,21,3,88,96,165,197,201,6
4,240
49212 DATA 40,205,127,193,240,35,141,12
7,193,201
49222 DATA 4,208,8,173,124,193,73,255,1
41,124
49232 DATA 193,201,5,208,8,173,125,193,
73,255
49242 DATA 141,125,193,201,6,240,31,201
,3,240
49252 DATA 3,76,49,234,169,70,32,153,19
2,141
49262 DATA 122,193,141,128,193,165,21,1
41,123,193
49272 DATA 141,129,193,162,255,108,0,3,
169,91
49282 DATA 32,153,192,141,126,193,173,1
28,193,141
49292 DATA 122,193,173,129,193,141,129,
193,162,255
49302 DATA 108,0,3,160,193,32,30,171,32
,96
49312 DATA 165,134,122,132,123,32,115,0
,170,240
49322 DATA 243,162,255,134,58,144,1,96,
32,107
49332 DATA 169,169,0,133,198,165,20,96,
32,87
49342 DATA 241,8,133,251,201,13,240,4,1
65,251
49352 DATA 40,96,173,124,193,240,247,16
9,13,32
49362 DATA 210,255,165,214,141,130,193,
206,130,193
49372 DATA 134,252,132,253,173,123,193,
174,122,193
49382 DATA 133,98,134.99.142,128,193,14
1,129,193
49392 DATA 162,144,56,32,73,188,32,223,
189,162
49402 DATA 0,189,0,1,240,7,157,119,2,23
2
49412 DATA 76,251,192,169.32,157,119,2,
232,173
49422 DATA 125,193,208,29,134,198,173,1
22,193,24
49432 DATA 109,126,193,141.122,193,144,
3,238,123
49442 DATA 193,166,252.164.253,173,130,
193,133,214
49452 DATA 76,198,192,160,0,185,65,193,
240,8
49462 DATA 200,157,119,2,232,76,49,193,
76,18
49472 DATA 193,68,65,84,65,0,13,67,79,7
7
49482 DATA 73,69,78.90,79,32,69,78,32,7
6
49492 DATA 73,78,69,65,63,32,0,13,81,85
49502 DATA 69,32,86,65,76.79,82,32,73,7
8
49512 DATA 67,82,69,77,69,78,84,79,40,4
8
49522 DATA 45,50,53,53,41,63,32,0,0,0
49532 DATA 0,0,10,64,0,0
```




CONTADOR DE BLOQUES LIBRES:

Comp.: Commodore 64.
Conf.: 64K con disquetera Commodore 1541.
Clas.: Utilitario.



Este rápido utilitario en lenguaje de máquina ha sido preparado para residir en memoria junto con un programa BASIC y tiene como objetivo informar al usuario la cantidad de bloques existentes en el disquette alojado en el drive con número de dispositivo B. Este contador será particularmente útil para decidir si existe espacio suficiente para grabar (SAVE) el programa BASIC que se encuentra en memoria, evitando el trabajo de cambiar disquettes o cargar el directorio que, si no se cuenta con el DOS 5.1 (ver disquette DEMO del 1541), borrará el programa BASIC. Una vez cargado en memoria, el utilitario puede ser llamado tecleando en modo directo, SYS 4915. ATENCION: antes de cargar este utilitario, verifiquemos que el área RAM que comienza en 49152 no se halla ocupada por algún otro programa. De residir en ella un utilitario tal como el numerador automático, el C-64 sufrirá un "crash" irrecuperable (habrá que reencenderlo para que vuelva a comportarse correctamente).

```
100 REM ********************************
105 REM *                              *
110 REM *  CONTADOR DE BLOQUES LIBRES  *
115 REM *                              *
120 REM * REVISTA K64, AGOSTO DE 1985. *
125 REM *                              *
130 REM ********************************
140 PRINT"{CLR}":POKE53281,0
145 PRINT"ESTE PROGRAMA CARGA UNA RUTINA
EN "
150 PRINT"LENGUAJE DE MAQUINA EN MEMORIA.
"
160 FORX=49152TO49307:READA:B=B+A
165 NEXT:IFB<>20757THEN220
170 RESTORE
175 FORX=49152TO49307:READA
180 POKEX,A:NEXT:PRINT
185 PRINT"LA RUTINA SE EJECUTARA"
190 PRINT"TECLEANDO 'SYS 49152'"
195 PRINT"Y PRESIONANDO <RETURN>. A CONTI
NUACION"
200 PRINT"CONTARA LA CANTIDAD DE BLOQUES
LIBRES"
205 PRINT"DISPONIBLES EN EL DISCO DEL DRI
VE 8."
210 PRINT"{C/DN}{C/DN}VERIFIQUE QUE LA DI
SQUETERA ESTE"
215 PRINT"ENCENDIDA!!":END
220 PRINTCHR$(28)
225 PRINT"{CLR}EXISTE UN ERROR EN UNA"
230 PRINT"SENTENCIA 'DATA'."
235 PRINT"{C/DN}VERIFIQUE EL LISTADO Y"
240 PRINT"CORRIJA VALORES INCORRECTOS."
245 END
250 DATA 169,2,162,8,160,2,32,186,255
255 DATA 169,1,162,137,160,192,32,189
260 DATA 255,32,192,255,32,68,229,162
265 DATA 2,32,198,255,169,0,141,136
270 DATA 192,169,2,141,134,192,32,228
275 DATA 255,32,228,255,32,228,255,141
280 DATA 135,192,32,228,255,32,228,255
285 DATA 32,228,255,32,228,255,174,134
290 DATA 192,224,18,240,18,24,109,135
295 DATA 192,141,135,192,173,136,192
300 DATA 105,0,141,136,192,174,134,192
305 DATA 232,142,134,192,224,36,208,211
310 DATA 169,141,32,210,255,32,210,255
315 DATA 174,135,192,173,136,192,32,205
320 DATA 189,169,32,32,210,255,169,138
325 DATA 160,192,32,30,171,32,204,255
330 DATA 169,2,32,195,255,96,0,0
335 DATA 0,36,66,76,79,81,85,69,83,32
340 DATA 83,73,78,32,85,83,65,82,46,0
```

AHORRO DE MEMORIA DE SU 1000/1500

(Parte III)

Continuando con el número anterior, ofrecemos la tercera y última parte de este interesante tema.

¿CUANDO CONSUMIMOS 24 BYTES?:

10 IF A=1 THEN GOTO 9 pero, 10 IF A=1 THEN GOTO 10 consume 25 bytes.
El uso de "<=","<",">" y "<>" en líneas como ésta, también dan un consumo idéntico.
10 IF A=1 OR B < 2 THEN GOTO 9 consume 34 bytes, de modo que "OR B < 2" insume 10 bytes.
10 IF NOT A=1 THEN GOTO 9 consume 25 bytes poi lo que NOT solamente consume 1 byte.
10 IF A=1 THEN GOSUB 9 también consume 24 bytes.

IMPRIMIENDO ORDENADAMENTE

Existen dos funciones asociadas a la sentencia PRINT que imprimen en un lugar determinado de la pantalla, estas funciones son AT y TAB, veamos cada una de ellas:

PRINT AT:

10 PRINT AT 1,1;"A" consume 26 bytes, de modo que "AT 1,1;" insume 17 bytes, en gran parte por utilizar los números, en cambio si escribimos 10 PRINT AT L,L;"A" se consumirán solamente 14 bytes.

TAB:

10 PRINT TAB 1;"A" consume 18 bytes, lo cual hace que "TAB 1;" tome 9 bytes.
10 PRINT TAB X;"A" en este caso sólo consume 12 bytes.

COMPARACION ENTRE PRINT AT Y SENTENCIAS PRINT VACIAS

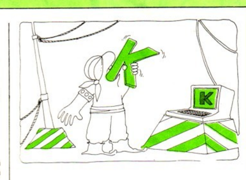
Si tenemos:
10 PRINT AT 4,1;"A" consume como dijimos 26 bytes, pero

```
10 PRINT
15 PRINT
20 PRINT        consume 27 bytes
25 PRINT "A"
```

Como se puede apreciar, es muy importante utilizar sentencias PRINT vacías para dejar una o dos líneas en blanco, pero para tres o más, es económico la utilización del PRINT AT.
También podemos tener el caso 10 PRINT AT 2,5;"A" con 26 bytes de consumo, reemplazar por:

```
10 PRINT
15 PRINT TAB 5;"A"
```

donde el consumo es de 24 bytes
Si se emplean nombres de variables en vez de números, entonces el imprimir ordenadamente puede ser más conveniente y efectivo, así por ejemplo:
10 PRINT AT X,Y;"A" solamente consumirá 14 bytes.



SUBCADENAS

10 PRINT A$ tiene un consumo de 8 bytes, en cambio 10 PRINT A$ (TO 9) consume 18 bytes de modo que "(TO 9)" insume 10 bytes.
10 PRINT A$ (TO 10) nos consume 19 bytes debido al caracter extra.
10 PRINT A$ (1 TO 9) consume 25 bytes.
10 PRINT A$ (1 TO) también consume 18 bytes.
Aquí otra vez, podemos ahorrar memoria utilizando nombre de variables en lugar de números.
10 PRINT A$ (A TO B) consume solamente 13 bytes, comparándolo con los 25 bytes.

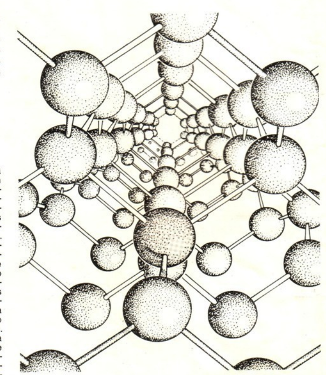
RESUMEN

Una vez analizados estos consumos es importante tener a mano una pequeña subrutina, de modo tal, que cuando se está programando nos indique la longitud del programa, es decir, la cantidad de bytes utilizados.
La subrutina que se debe agregar es:
9999 PRINT "LONGITUD DEL PROGRAMA –"; PEEK 16396+256*PEEK 16397-16596; "–BYTES"
donde – indica un espacio.
Si esta subrutina se ejecuta, el resultado es "0 BYTES". La manera de trabajar con ella es la siguiente:
Una vez que se ingresó un programa o en un estado intermedio, antes de correrlo hacemos un GOTO 9999, y en pantalla aparecerá la cantidad de bytes insumidos.
Les deseamos que tengan muchos éxitos.

Ing. JULIO JOSE PUTRUELE
Ing. MIGUEL ANGEL MAUBRO

AVANZA LA TECNOLOGIA DE LOS SEMICONDUCTORES

Progresos en la tecnología de los semiconductores se están logrando en los laboratorios de IBM, que auguran computadoras más veloces, más reducidas y de menor consumo.



Cristal de silicio semiconductor. Su perfecta irregularidad está interrumpida por un átomo de "impureza".

No es ningún secreto que la tecnología de la computación ha avanzado a un paso enérgico desde la aparición de los primeros modelos. Hoy se puede tener sobre el escritorio mucha más potencia informática que con aquellos "monstruos" de hace unos años. Los modelos de los próximos tiempos serán aún más pequeños, potentes y baratos qye los de este año. Y de las máquinas de dentro de diez, deberemos estar preparados para ver milagros. Pero para los investigadores del SST, el departamento de ciencia y tecnología del semiconductor, el progreso no es cosa de milagros instantáneos. En realidad es producto de una larga trayectoria y gran cantidad de conocimientos científicos acumulados. La tecnología computacional del futuro es la mayor preocupación de los hombres y mujeres que trabajan en el SST y mucho de lo que se vea en la próxima década, será consecuencia del trabajo que hoy están realizando.

"Nosotros somos los responsables de poseer los programas más avanzados de investigación en la ciencia del semiconductor", dice John Armstrong, director del SST, y vicepresidente del Departamento de Investigación en Lógica y Memoria. "Es una gran responsabilidad. IBM tiene miles de personas trabajando en la tecnología del semiconductor y en el encapsulado, por eso debemos conocer a fondo su trabajo". Existe un programa que encara el SST y otras secciones para investigar en conjunto qué es lo que se necesita. El Advanced Packaging Technology Laboratory (APTL) y el Advanced Silicon Technology Laboratory (ASTL) son dos buenos ejemplos de ellos. "El trabajar en un programa conjunto hace que se puedan explotar a fondo las fuentes en tecnología avanzada de IBM", afirmó Armstrong. "Además ayuda a la transferencia de tecnología".



CONCURSO TRIMESTRAL



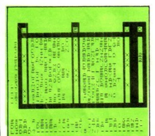
PRODE

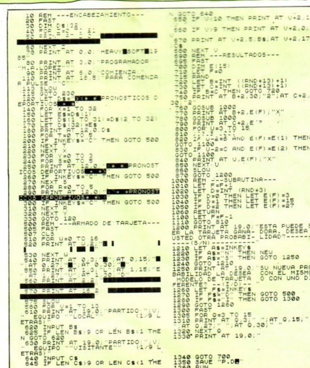
Comp: CZ1000/1500 TK83/85
Conf: 16 K
Clas: Entretenimiento
Autor: Manuel A. López

LISTA DE VARIABLES

Z LOOP: Tiñe la pantalla de negro.
Y LOOP: Controla los cambios del título.
X LOOP: Controla el movimiento del título.
W LOOP: Dibuja las columnas de la tarjeta.
V LOOP: Permite el ingreso de los 13 partidos.
U LOOP: Selecciona los 13 resultados.
T LOOP: Permite el comienzo del juego.
S LOOP: Imprime durante un momento el título en video normal.
R LOOP: Imprime durante un momento el título en video inverso.
Q LOOP: Borrado de cruces.

PANTALLA





```
  10 REM ---ENCABEZAMIENTO---
  20 FAST
  30 DIM D$(32)
  40 FOR Z=0 TO 21
  50 PRINT AT Z,0;"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"
  60 NEXT Z
  70 PRINT AT 0,0;"HEAVY■SOFT■19
85"
  80 PRINT AT 3,0;"PROGRAMADOR
"M.A.LOPEZ""
  90 PRINT AT 6,0;"COMIENZA:"
 110 PRINT AT 18,5;"PARA COMENZA
R PULSE ""C"""
 113 SLOW
 115 GOTO 230
 120 LET D$=" ><+ + PRONOSTICOS D
EPORTIVOS + +"
 140 FOR X=1 TO 32
 150 LET E$=D$(1)
 160 LET D$(1 TO 31)=D$(2 TO 32)
 170 LET D$(32)=E$
 180 PRINT AT 12,0;D$
 190 FOR T=0 TO 5
 200 IF INKEY$="C" THEN GOTO 500
 210 NEXT T
 220 NEXT X
 230 FOR Y=0 TO 3
 240 FOR S=0 TO 5
 250 PRINT AT 12,0;" + + PRONOST
ICOS DEPORTIVOS + +>"
 260 IF INKEY$="C" THEN GOTO 500
 270 NEXT S
 280 FOR R=0 TO 5
 290 PRINT AT 12,0;" + +PRONOST
ICOS DEPORTIVOS+ + <"
 300 IF INKEY$="C" THEN GOTO 500
 310 NEXT R
 320 NEXT Y
 330 GOTO 120
 500 REM ---ARMADO DE TARJETA---
 505 FAST
 507 CLS
 510 FOR W=0 TO 16
 520 PRINT AT W,2;"■▌          ▐
▌        ▐■■ ■"
 530 NEXT W
 540 PRINT AT 0,3;"■";AT 0,15;"■
";AT 0,27;"■";AT 0,30;"■"
 550 PRINT AT 1,3;"L";AT 1,15;"E
";AT 1,27;"V";AT 1,30;"D"
 560 PRINT AT 2,2;"■■■■■■■■■■■■■■
■■■■■■■■■■■■■■■"
 570 PRINT AT 16,2;"■■■■■■■■■■■■■
■■■■■■■■■■■■■■■"
 575 SLOW
 580 FOR V=1 TO 13
 610 PRINT AT 19,0;"PARTIDO:";V;
"  EQUIPO ""LOCAL""          (1/9 L
ETRAS)"
 620 INPUT B$
 625 IF LEN B$>9 OR LEN B$<1 THE
N GOTO 620
 630 PRINT AT 19,0;"PARTIDO:";V;
"  EQUIPO ""VISITANTE""   (1/9 L
ETRAS)"
 640 INPUT C$
 645 IF LEN C$>9 OR LEN C$<1 THE
N GOTO 640
 650 IF V<10 THEN PRINT AT V+2,1
;V
 660 IF V>9 THEN PRINT AT V+2,0;
V
 670 PRINT AT V+2,5;B$;AT V+2,17
;C$
 680 NEXT V
 690 REM ---RESULTADOS---
 695 FAST
 700 DIM E(15)
 710 LET F=0
 715 RAND
 720 LET B=INT ((RND*13)+1)
 730 LET C=INT ((RND*13)+1)
 740 IF B=C THEN GOTO 720
 750 PRINT AT B+2,30;"2";AT C+2,
30;"2"
 760 GOSUB 1000
 770 PRINT AT B+2,E(F);"X"
 780 GOSUB 1000
 790 PRINT AT C+2,E[illegible]
 800 FOR U=3 TO 15
 810 [illegible]
 820 IF U-2=B AND E(F)=E(1) THEN
 GOTO 1100
 830 IF U-2=C AND E(F)=E(2) THEN
 GOTO 1100
 840 PRINT AT U,E(F);"X"
 850 NEXT U
 860 SLOW
 870 GOTO 1200
1000 REM ---SUBRUTINA---
1010 LET F=F+1
1020 LET D=INT (RND*3)
1030 IF D=0 THEN LET E(F)=3
1040 IF D=1 THEN LET E(F)=15
1050 IF D=2 THEN LET E(F)=27
1060 RETURN
1100 LET F=F-1
1110 GOTO 810
1200 PRINT AT 19,0;"ESTA PUEDE S
ER LA TARJETA GANA- DORA. DESEA
USTED OTRA PROBABI- LIDAD?------
----(S/N)-----------"
1210 LET A$=INKEY$
1220 IF A$="N" THEN NEW
1230 IF A$="S" THEN GOTO 1250
1240 GOTO 1210
1250 PRINT AT 19,0;"SU NUEVA PRO
BABILIDAD LA DESEA  CON EL MISMO
 MODELO DE TARJETA  O CON UNO DI
FERENTE? --(I/D)-- "
1260 LET F$=INKEY$
1270 IF F$="D" THEN GOTO 500
1280 IF F$="I" THEN GOTO 1300
1290 GOTO 1260
1300 FAST
1305 FOR Q=3 TO 15
1310 PRINT AT Q,3;" ";AT Q,15;"
";AT Q,27;" ";AT Q,30;" "
1320 NEXT Q
1330 PRINT AT 19,0;"


1340 GOTO 700
1350 SAVE "P.D■"
1360 RUN
```


K64

A$ ALFANUMERICA: Ingresa la contestación a una nueva probabilidad.
B$ ALFANUMERICAS: Ingresa el nombre del equipo local.
C$ ALFANUMERICAS: Ingresa el nombre del equipo visitante.
D$ ALFANUMERICAS (Dimensionada): Guarda el título.
E$ ALFANUMERICAS: Guarda el Primer caracter de D$.
F$ ALFANUMERICAS: Ingresa la contestación al nuevo tipo de tarjeta.
F: Controla el número de resultado.
E (Dimensionada): Guarda los resultados.
TIPO DE PROGRAMA: Juego de Azar.
CAPACIDAD DE MEMORIA: 2,9 K.
MAQUINAS EN QUE PUEDE SER CORRIDO: TK 85 y similares cuya configuración mínima de memoria sea de 16 K.
INSTRUCCIONES: Sólo hay que obedecer al programa.
TIPO DE CINTA: El programa fue grabado con una TK 85 en una cinta del tipo NORMAL sin ningún tipo de ecualización especial.
NUMERO DE CUENTAVUELTAS: 063.

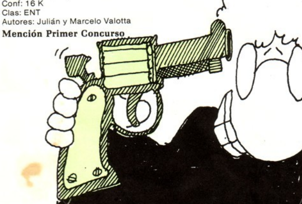
COMENTARIO DE LINEAS:

10-113: Arman la presentación del juego.
115-330: Generan el movimiento y el cambio de video del título.
500-575: Arman la tarjeta.
580-680: Permiten la entrada de los nombres de los equipos.
690-790: Seleccionan los dobles al azar y los imprimen.
800-860: Eligen los resultados posibles, los compara con los resultados de los dobles ya elegidos, si el partido es el mismo al igual que el resultado, elige un nuevo resultado, de lo contrario imprime.
700: Envía ejecutar las preguntas.
1000-1110: Subrutina de selección de resultados.
1200-1290: Pregunta por nuevas probabilidades y tipo de tarjeta.
1300-1320: Borrado de cruces.
1330: Borrado de pregunta.
1340: Manda una nueva selección de resultados.
1350: Graba el programa.
1360: Corre el programa.

RULETA RUSA

Comp: TK 83/85, CZ 1000/1500
Conf: 16 K
Clas: ENT
Autores: Julián y Marcelo Valotta

Mención Primer Concurso



El Programa, en el cual está presente el azar, consta de gráficos en una cadena de la forma A$ (por ej.) De esta menera, los revólveres y la tumba que aparecen durante el juego, están incluidos dentro de las correspondientes cadenas.
En el listado del programa, las variables A$, B$, y C$ no están definidas, por lo tanto una vez que se haya terminado de tipear el programa y antes de ejecutarlo, hay que darle el valor a cada una de estas variables de la siguiente manera: LET nombre de variable "contenido", sin número de línea.

Por este motivo nunca hay que hacer correr el programa con el comando RUN. En cambio hay que usar GOTO 1.

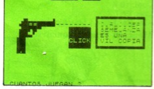
COMO FUNCIONA EL JUEGO:

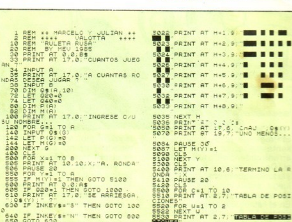
Al empezar, pregunta el número de jugadores, el número de rondas y el nombre de cada uno de los participantes.
Por turno los jugadores se deberán arriesgar -o no- a "apretar" el gatillo de la pistola que aparece en pantalla.
A la pregunta: "se arriesga", uno puede responder presionando "S" (si) ó "n" (no); si contestamos afirmativamente, nos preguntará cuál es el número que elegimos (de 1 a 6), simulando así el tambor de la pistola de 6 tiros.
Presionaremos a continuación el número que sea de nuestro agrado y es aquí donde se hace presente el azar.
La computadora elige entonces -a su vez- un número del 1 al 6, y si este coincide con el elegido por nosotros... ¡PUM!... uno menos en el juego (aparecerá una tumba y la computadora anunciará nuestro final).
Si a la pregunta "se arriesga..." se contesta negativamente (pulsando N), la computadora nos tildará de cobardes.
A continuación se repiten los pasos anteriores con el siguiente participante; así hasta terminar la primer ronda.
Se trata de obtener la máxima cantidad de puntos, consiguiéndose estos puntos, al oprimir el disparador y no ser víctimas de la imaginaria bola alojada en el tambor de la pistola: un disparo fallido un punto.
Si por casualidad sucumbimos en el intento, la computadora nos pasará por alto en las siguientes ron-

das: aunque puede darse el caso de que ningún otro participante, al terminar el juego, pueda alcanzarlo en la tabla de posiciones hasta el final, quedando así como el vencedor.
Alguien puede llegar a tener la posibilidad de probar suerte dos o más veces en una misma ronda, aumentando así el puntaje.
Al final de cada ronda se muestra la tabla de posiciones, donde aparecen todos los participantes ordenados de acuerdo a su cantidad de puntos.

Y se repiten todos los pasos anteriores, hasta concluir el número de rondas elegido al principio.

PANTALLA





```
   1 REM ** MARCELO Y JULIAN **
   2 REM ****    VALOTTA    ****
  10 REM "RULETA RUSA"
  20 REM  BY MEV 1985
  30 PRINT AT 0,0;B$
  33 PRINT AT 17,0;"CUANTOS JUEG
AN ?"
  34 INPUT A
  35 PRINT AT 17,0;"A CUANTAS RO
NDAS DESEA JUGAR ? "
  38 INPUT B
  70 DIM O$(A,10)
  72 LET O20=0
  74 LET O40=0
  80 DIM P(A)
  82 DIM M(A)
 100 PRINT AT 17,0;"INGRESE C/U
SU NOMBRE..."
 120 FOR G=1 TO A
 140 INPUT O$(G)
 142 LET P(G)=0
 144 LET M(G)=0
 200 NEXT G
 400 CLS
 500 FOR X=1 TO B
 505 PRINT AT 10,10;X;"A. RONDA"
 506 PAUSE 20
 550 FOR Y=1 TO A
 555 IF M(Y)=1 THEN GOTO 5100
 600 PRINT AT 0,0;A$
 605 IF O20=1 THEN GOTO 1000
 610 PRINT AT 17,0;"SE ARRIESGA,
 ";O$(Y);" ?"
 630 IF INKEY$="S" THEN GOTO 100
0
 640 IF INKEY$="N" THEN GOTO 800
 650 GOTO 630


 970 PRINT AT 20,0;''COBARDE''
 980 PAUSE 50
 990 CLS
 992 GOTO 5100
1000 PRINT AT 18,0;"ELIJA UNA TE
CLA DE 1 A 6"
1020 LET R$=INKEY$
1040 IF CODE R$>28 AND CODE R$<3
5 THEN GOTO 2000
1050 GOTO 1020
2000 LET O=INT (RND*6+1)
2020 IF CODE R$=O+28 THEN GOTO 5
000
2050 CLS
2070 PRINT AT 8,10;"CLICK..."
2080 PRINT AT 10,3;"FELICIDADES,
 GANO 1 PTO"
2085 LET P(Y)=P(Y)+1
2090 PRINT TAB 3;"YA COSECHO ";P
(Y);" PTS"
2100 LET O20=INT (RND*3+1)
2200 IF O20<>1 THEN PAUSE 30
2220 IF O20<>1 THEN GOTO 5100
2300 PRINT AT 13,0;"SE QUIERE AR
RIESGAR DE NUEVO ?"
2310 IF INKEY$="S" THEN GOTO 600
2311 IF INKEY$="N" THEN LET O20=
0
2320 IF INKEY$="N" THEN GOTO 509
0
2330 GOTO 2310
5000 CLS
5010 FOR H=10 TO 1 STEP -1
5020 PRINT AT H,9;"███ █ █ █   █
█"
5022 PRINT AT H+1,9;"███ █ █ ██
██ █"
5023 PRINT AT H+2,9;"███ █ █ █ █
 █ █"
5024 PRINT AT H+3,9;"███ █ █ █
 █ █"
5026 PRINT AT H+4,9;"█  █ █ █
 █ █"
5027 PRINT AT H+5,9;"█  █ █ █
 █ █"
5030 PRINT AT H+6,9;"█  ███ █
 █"
5032 PRINT AT H+7,9;"█  ███ █
 █ █"
5033 PRINT AT H+8,9;"
"
5035 NEXT H
5036 PRINT AT 0,0;C$
5050 PRINT AT 17,6;"CHAU ";O$(Y)
5070 PRINT AT 19,7;"UNO MENOS...
"
5084 PAUSE 30
5087 LET M(Y)=1
5090 CLS
5100 NEXT Y
5300 CLS
5400 PRINT AT 10,6;"TERMINO LA R
ONDA ";X
5410 PAUSE 20
5420 CLS
5500 FOR C=1 TO 10
5510 PRINT AT 2,7;"TABLA DE POSI
CIONES"
5520 FOR W=1 TO 2
5522 NEXT W
5530 PRINT AT 2,7;"TABLA DE POSI
CIONES"
5540 NEXT C
5542 PRINT
5600 LET E=0
5602 LET O40=O40+10
5603 FAST
5604 PRINT "POSIC.";" NOMBRE";TA
B 18;"PTS";"   CONDIC."
5606 PRINT "======";" ======";TA
B 18;"===";"   ======="
5608 PRINT
5620 FOR S=O40 TO 0 STEP -1
5700 FOR T=1 TO A
5720 IF M(T)=0 THEN LET R$="VIVO
"
5740 IF M(T)=1 THEN LET R$="MUER
TO"
6000 IF P(T)=S THEN LET E=E+1
6010 IF E=1 THEN LET O40=P(T)
6020 IF P(T)=S THEN PRINT E;TAB
7;O$(T);TAB 18;P(T);TAB 23;R$
6040 NEXT T
6050 NEXT S
6070 SLOW
7000 PAUSE 500
7500 CLS
8000 NEXT X
9000 PRINT
9010 PRINT
9012 PRINT "TERMINO EL SUPLICIO.
.."
9014 PRINT "FELICIDADES AL VENCE
DOR"
9016 PRINT TAB 10;"-PRESS ANY KE
Y-"
9020 PAUSE 9000
9030 GOTO 1
9100 SAVE "RULETA RUSA"
9120 GOTO 1
```

SALVAR VIDAS

Comp: CZ1000/1500 TK83/85
Conf: 16 K
Clas: Entretenimiento
Autor: Julio Moreno

PANTALLA

```
   1 REM  **** SALVAR VIDAS ****
   2 REM  **********************
   3 REM  *    ENVIADO  POR    *
   4 REM  **** JULIO MORENO ****
   5 PRINT AT 10,3;"BUSCA UNA LL
AVE T ENTRE (10?)    DISPONES DE
POCO TIEMPO SI       LO LOGRAS
PODRAS PASAR LA      CELDA █ Y
DEBES TOCAR LA       PERSONA TE
MUEVES CON LAS       TECLAS 5 6
7 8"
   8 PAUSE 600
  10 PRINT AT 0,0;"██████████████
██████████████████"
  15 PRINT AT 21,0;"█████████████
██████████████████"
  16 FOR A=1 TO 20
  17 PRINT AT A,1;"██████████████
████████████████"
  18 NEXT A
  20 FOR L=1 TO 20
  22 PRINT AT L,0;"█"
  23 NEXT L
  25 FOR U=1 TO 10
  26 PRINT AT 10,30;"█"
  27 PRINT AT 10,30;"█"
  29 NEXT A
  32 FOR H=0 TO 21
  33 PRINT AT H,27;"█"
  35 NEXT H
```






TORRES de HANOI

Comp: CZ1000/1500 TK83/85
Conf: 16K
Clas: Entretenimiento
Autor: Miguel García



Este es un programa de entretenimiento basado en el conocido juego de las Torres de Hanoi. Básicamente consiste en la traslación de una torre de una base a otra, con dos bases auxiliares, y teniendo en cuenta que toda plataforma debe ser colocada sobre una de tamaño superior.

Carece de instrucciones porque es sencillo y porque además el jugador puede ver cómo la máquina resuelve el problema planteado.

La codificación corresponde al lenguaje BASIC de la micro Sinclair ZX81.

La memoria requerida para su funcionamiento es de 16 K.

En la línea 20 se elige el nivel de la torre a trasladar con un tope de 9 plataformas. Desde la línea 30 hasta la 270 se cargan las variables y se presenta la torre. En la línea 280 se pide una opción:

a) "Vos (V)" para que el jugador pueda probar su capacidad. (600 a 770). Se piden los datos para el traslado, se validan dichos datos y si fueren adecuados se establece el cambio en la subrutina (900 a 970). El récord equivale a la cantidad de movimientos indispensables.

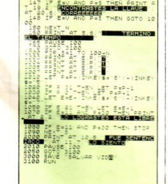
b) "Yo (Y)" para que la máquina resuelva el problema por sí misma. (315 a 590). Para resolverlo, almacena en un vector (V) los cambios necesarios. (320 a 510). Después ejecuta los cambios valiéndose de la subrutina mencionada. (520 a 560). Está claro que la máquina lo hace con la menor cantidad de movimientos.

c) "Otra torre (T)", si se desea una torre distinta por parecerle demasiado fácil o demasiado difícil.

en todos los casos se vuelve a las opciones de la línea 280.

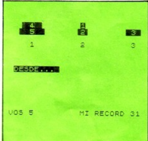
para dar una idea de lo que significa resolver una torre de Hanoi, se definen los movimientos indispensables como 2" - 1 con igual a la altura de la torre. Eso significa que resolver una torre de nivel 9 equivale a 511 movimientos como mínimo; esto supone para el jugador entrar 1022 datos.

Aunque el lenguaje BASIC no es recursivo puede, de vez en cuando, resolver problemas de recursividad.



```
  50 LET E=20
  52 LET P=1
  54 FOR D=1 TO 10
  56 LET V=INT (RND+19+1)
  58 LET S=INT (RND+20+1)
  68 PRINT AT V,S;"▓"
  69 NEXT D
  71 FOR N=1 TO 100
  72 PRINT AT E,P;"▓"
  73 PRINT AT E,P;"▓"
  74 PRINT AT E,P;"▓"
  75 PRINT AT E,P;"▓"
 100 LET P=P+(INKEY$="8")-(INKEY
$="5")
 110 IF P>26 THEN LET P=P-1
 120 IF P<1 THEN LET P=P+1
 130 LET E=E+(INKEY$="6")-(INKEY
$="7")
 140 IF E>20 THEN LET E=E-1
 145 IF E<1 THEN LET E=E+1
 147 IF E=V AND P=S THEN PRINT A
T 19,2;"ENCONTRASTES LA LLAVE ";
AT 20,6;"CORREEEEEE"
 148 IF E=V AND P=S THEN GOTO 10
00
 150 NEXT N
 160 PRINT AT 5,6;"      TERMINO
EL TIEMPO      "
 165 PAUSE 100
 170 GOTO 3100
1000 FOR S=1 TO 100-N
1003 PRINT AT E,P;"▓"
1004 PRINT AT E,P;"▓"
1005 PRINT AT E,P;"▓"
1007 PRINT AT E,P;"▓"
1010 LET P=P+(INKEY$="8")-(INKEY
$="5")
1020 IF P>31 THEN LET P=P-1
1030 IF P<0 THEN LET P=P+1
1040 LET E=E+(INKEY$="6")-(INKEY
$="7")
1050 IF E>21 THEN LET E=E-1
1070 IF E=11 AND P=30 THEN PRINT
 AT 2,2;"LO LOGRASTES ESTA LIBRE
    "
1080 IF E=11 AND P=30 THEN STOP
1090 NEXT S
2000 PRINT AT 10,5;" FUE SENTENC
IADO ";AT 11,7;"LO SIENTO"
2010 PAUSE 100
2050 GOTO 3100
3000 SAVE "SALVAR VID▓"
3100 RUN
```

PANTALLA



PANTALLA







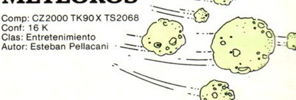
CONCURSO TRIMESTRAL



```
  10 SLOW
  20 PRINT "     NIVEL  DE  LA  TO
RRE          "
  30 INPUT N
  40 IF N>9 OR N<1 THEN GOTO 30
  50 LET L=2**N-1
  60 LET M=N+1
  70 DIM Y(3)
  80 DIM Q$(3,9,9)
  90 CLS
  95 LET W=0
 100 LET Y(1)=1
 110 LET Y(2)=M
 120 LET Y(3)=M
 130 LET Q$(1,1)="         1         "
 140 LET Q$(1,2)="         2         "
 150 LET Q$(1,3)="         3         "
 160 LET Q$(1,4)="         4         "
 170 LET Q$(1,5)="         5         "
 180 LET Q$(1,6)="         6         "
 190 LET Q$(1,7)="         7         "
 200 LET Q$(1,8)="         8         "
 210 LET Q$(1,9)="         9         "
 220 FOR K=1 TO N
 230 PRINT AT K,1;Q$(1,K)
 240 NEXT K
 250 PRINT AT N+2,5;"1"
 260 PRINT AT N+2,15;"2"
 270 PRINT AT N+2,25;"3"
 280 PRINT AT 18,0;"  VOS(V)   YO
(Y)   OTRA TORRE (T) "
 290 IF INKEY$="T" THEN RUN
 300 IF INKEY$="V" THEN GOTO 600
 310 IF INKEY$<>"Y" THEN GOTO 29
0
 315 FAST
 320 LET T=2*(2**N-1)
 330 DIM V(T)
 340 FOR I=1 TO N
 350 LET J=2**I-1
 360 LET V(J)=1
 370 IF I/2-INT (I/2)=0 THEN GOT
O 400
 380 LET V(J+1)=2
 390 GOTO 410
 400 LET V(J+1)=3
 410 LET X=0
 420 FOR J=2**I+1 TO 2**(I+1)-2
 430 LET X=X+1
 440 IF I/2-INT (I/2)<>0 THEN GO
TO 470
 450 LET V(J)=V(X)+1
 460 GOTO 490
 470 LET V(J)=V(X)+2
 480 IF V(J)=5 THEN LET V(J)=2
 490 IF V(J)=4 THEN LET V(J)=1
 500 NEXT J
 510 NEXT I
 515 SLOW
 520 FOR I=1 TO T STEP 2
 530 LET A=V(I)
 540 LET B=V(I+1)
 550 GOSUB 900
 560 NEXT I
 570 PRINT AT 18,0;" LO HE CONSE
GUIDO EN ";W;" INTENTOS"
 580 PAUSE 200
 590 GOTO 70
 600 PRINT AT 18,0;" VOS ";W;"
      ";AT 18,15;"MI RECORD ";L
;"
 "
 610 PRINT AT N+6,2;"DESDE...
 "
 620 INPUT A
 630 PRINT AT N+6,10;"HASTA..."
 640 INPUT B
 650 IF A<1 OR A>3 THEN GOTO 610
 660 IF B<1 OR B>3 THEN GOTO 610
 670 IF Y(A)<>M THEN GOTO 690
 680 GOTO 610
 690 IF Y(B)=M THEN GOTO 710
 700 IF CODE (Q$(A,Y(A),5))>CODE
 (Q$(B,Y(B),5)) THEN GOTO 610
 710 GOSUB 900
 720 PRINT AT 18,5;W
 730 IF Y(2)=1 OR Y(3)=1 THEN GO
TO 750
 740 GOTO 610
 750 PRINT AT 20,0;"  COMPARE LO
S RESULTADOS           "
 760 PAUSE 200
 770 GOTO 70
 900 REM *---*  SUBRUTINA  ----*
 910 PRINT AT Y(A),A*10-9;"
 "
 920 LET Y(B)=Y(B)-1
 930 PRINT AT Y(B),B*10-9;Q$(A,Y
(A))
 940 LET Q$(B,Y(B))=Q$(A,Y(A))
 950 LET Y(A)=Y(A)+1
 960 LET W=W+1
 970 RETURN
```

METEOROS

Comp: CZ2000 TK90 X TS2068
Conf: 16 K
Clas: Entretenimiento
Autor: Esteban Pellacani



```
   1 REM ************************
   2 REM *                      *
   3 REM * HECHO EN QUILMES POR *
   4 REM *                      *
   5 REM *  ESTEBAN PELLACANI   *
   6 REM *                      *
   7 REM ************************
  10 CLS
  15 PAUSE 150
  20 GOSUB 200
  30 FOR i=1 TO 4: PRINT AT b(i)
,c(i);" ";: NEXT i
  40 LET v=v+(INKEY$="0" AND v<2
3)-(INKEY$="1" AND v>3)
  50 PRINT AT 20,v; INK 6;v$;
  60 FOR i=1 TO 4
  70 IF b(i)<20 THEN GOTO 120
  80 LET b(i)=4:: LET c(i)=v-3+I
NT (RND*11)
  90 LET p=p+1: PRINT AT 0,8;p;
 100 IF p=25 THEN LET s=2
 105 IF p=60 THEN GOTO 500
 110 GOTO 130
 120 LET b(i)=b(i)+s
 130 SOUND .01,17: PRINT AT b(i)
,c(i); INK i;s$;
 140 IF b(i)<20 THEN GOTO 160
 150 IF (c(i)-v<5 AND c(i)-v>0)
THEN SOUND 2,20: GOTO 510
 160 NEXT i
 170 GOTO 30
 200 PAPER 5: CLS
 210 LET v$=" "+CHR$ 134+CHR$ 14
0+CHR$ 140+CHR$ 137+" "
 220 LET s$=CHR$ 140
 230 LET s=1: LET p=0: LET v=13
 240 DIM b(4): DIM c(4)
 250 FOR i=1 TO 4: READ b(i): RE
AD c(i): NEXT i
 260 PRINT AT 0,0;"PUNTOS: 0";
 270 RETURN
 280 DATA 16,4,12,8,4,16,8,20
 500 FOR g=1 TO 50: SOUND .01,g:
 NEXT g: FOR w=1 TO 4: FOR x=1 T
O 7: SOUND .01,x+15: BORDER x: P
APER x: CLS : NEXT x: NEXT w
 510 CLS : PRINT ; PAPER 4; INK
7;AT 11,8;"FELICITACIONES"; PAPE
R 1;AT 19,6;"Muy buen puntaje: "
; PAPER 7; INK 0;p
 600 INPUT ; PAPER 4; INK 1;"Qui
ere jugar nuevamente (S/N)";q$
 610 IF q$="s" OR q$="S" THEN RU
N
 620 IF q$=" " THEN STOP
1000 STOP
9999 SAVE "METEOROS" LINE 10
```



EDUCAR

Comp: CZ2000 TK 90 X TS 2068
Conf: 16 K
Clas: Educativo
Autor: Carlos E. Silva



El programa comienza pidiendo el nombre del usuario, y e allí en adelante se dirigirá a éste por su nombre. Si se desea modificar alguno de los temas, se debe ingresar en vez del nombre la Palabra CLAVE. Si se desea modificar todos los temas, se debe ingresar CLAVE TOTAL. Para archivar los nuevos datos se debe ingresar ARCHIVA. En todos los casos se deben seguir las instrucciones de la Pantalla. La Pregunta y la opción pueden tener hasta 13 caracteres.

El título de cada tema puede tener hasta 32 caracteres. Pero el menú queda poco estético si tiene más de 25.

PANTALLA

```
Pulse una tecla para seguir
   A CAPITALES DE AFRICA
   B CAPITALES DE AMERICA
   C CAPITALES DE EUROPA
   D CAPITALES DE ASIA
   E CIUDADES DE AMERICA
   F CIUDADES DEL MUNDO
   G CAPITALES PROVINCIALES
   H CIUDADES ARGENTINAS
   I CHACO - CAP. DEPTOS. I
   J CHACO - CAP. DEPTOS. II
```

PANTALLA

```
           CAPITALES DE AFRICA
A LIBIA              0 LAGOS
B EGIPTO             1 MOGADISCIO
C NIGERIA            2 KAMPALA
D ZAIRE              3 TRIPOLI
E ANGOLA             4 KHARTUM
F SOMALIA            5 EL CAIRO
G ETIOPIA            6 LUANDA
H TANZANIA           7 KINSHASA
I SUDAN              8 MAPUTO
J MOZAMBIQUE         9 ADDIS ADEBA
DESEA INSTRUCCIONES? S=SI
```

PANTALLA

```
          CIUDADES DEL MUNDO
A BARCELONA          0 RUSIA
B AMSTERDAN          1 TURQUIA
C VENECIA            2 MEXICO
D FILADELFIA         3 ESPAÑA
E LENINGRADO         4 INDIA
F KIOTO              5 INGLATERRA
G GUADALAJARA        6 ITALIA
H CAMBRIDGE          7 EE. UU.
I ESTAMBUL           8 JAPON
J CALCUTA            9 HOLANDA
```

```
[illegible] POKE 23[illegible]
09,35: POKE 23[illegible]
  15 CLS : INK 5: FOR Z=10 TO 30
 STEP 5: SOUND .05,Z: NEXT Z: GO
SUB 8000: FOR Z=1 TO 20: SOUND .
03,Z: NEXT Z: PAUSE 30
  20 INK 7: GOSUB 5200
  25 GOSUB 5250: IF CV=2 THEN GO
TO 40
  27 IF CV=0 THEN GOTO 50
  30 GOSUB 6010
  32 GOTO 1
  35 GOTO 50
  40 GOSUB 6000
  45 GOTO 1
  50 GOSUB 6200
  60 GOSUB 6500
  70 GOSUB 1000
  80 GOSUB 2000
 100 GOSUB 6950: PRINT AT 21,2;"
CONTESTAS OTRO TEMA? N=NO": PAUS
E 0: IF INKEY$="N" THEN GOTO 680
0
 110 LET CON=1
 115 IF INS=0 THEN LET INS=1
 120 GOTO 50
1000 PRINT AT 21,2;"DESEA INSTRU
CCIONES? S=SI": PAUSE 0: LET E$=
INKEY$: PRINT AT 21,2;"
                    ": IF E$="S" THE
N LET Z$=N$+", "+I$(X): GOSUB 99
98: LET Z$=Z$+" ADELANTE....": G
OSUB 9985
1005 FOR Z=1 TO 10: SOUND .05,Z:
 NEXT Z
1010 LET A(X)=0
1020 FOR Y=1 TO 10
1025 INPUT "RESPUESTA DE ";: INV
ERSE 1;(CHR$ (Y+64)); INVERSE 0;
" EN NRO ",R(X,Y): IF R(X,Y)>9
THEN SOUND .05,50: GOTO 1025
1030 LET R(X,Y)=R(X,Y)+1: IF C(X
,Y)=R(X,Y) THEN FOR Z=1 TO 5: SO
UND .08,Z/10+5: NEXT Z: LET A(X)
=A(X)+1
1040 IF C(X,Y)<>R(X,Y) THEN FOR
Z=1 TO 5: SOUND .08,Z*10-5: NEXT
 Z
1050 NEXT Y: LET J$(X,2 TO )="-"
+STR$ A(X): RETURN
2000 CLS : GOSUB 6950: LET Z$="R
ESPUESTAS CORRECTAS": GOSUB 9900
2010 FOR Y=1 TO 10: GOSUB 6950:
PRINT INVERSE 1;CHR$ (Y+64); INV
ERSE 0;" ";P$(X,Y);TAB 17; INVER
SE 1;Y-1; INVERSE 0;" ";O$(X,C(X
,Y));
2020 IF C(X,Y)=R(X,Y) THEN PRINT
 AT 2+Y-1,17; FLASH 1;">"''
2030 IF C(X,Y)<>R(X,Y) THEN PRIN
T ''
2040 NEXT Y: FOR Z=40 TO 50: SOU
ND .05,Z: NEXT Z: LET Z$=N$: IF
CON=0 THEN LET Z$=Z$+", TUS ACIE
RTOS ESTAN INDICADOS CON "">"".
HAS OBTENIDO "+STR$ A(X)
2050 IF CON=1 THEN LET Z$=Z$+",
HAS OBTENIDO "+STR$ A(X)
2060 IF A(X)=1 THEN LET Z$=Z$+"
PUNTO."
2070 IF A(X)<>1 THEN LET Z$=Z$+"
 PUNTOS."
2090 GOSUB 9985: RETURN
5210 LET A$="ESTE PROGRAMA LE PE
RMITE REPASAR": GOSUB 5300
5215 LET A$="SUS CONOCIMIENTOS S
OBRE DISTIN-": GOSUB 5300
5220 LET A$="TOS TEMAS, QUE UD.
PUEDE SELEC-": GOSUB 5300
5225 LET A$="CIONAR A CONTINUACI
ON.": GOSUB 5300
5230 LET TI=4: LET Z$="Escrito p
or CARLOS EMILIO SILVA - Uruguay
 672 - 3509-Gral. San Martin - C
HACO - Mayo de 1985": GOSUB 9985
5240 FOR Z=20 TO 40: SOUND .03,Z
: NEXT Z: PRINT AT 21,0; INVERSE
 1;"   Pulse una tecla para segu
ir   ": PAUSE 0: RETURN
5250 CLS : LET INS=0: LET CON=0:
 LET CV=0: INPUT "INGRESE SU NOM
BRE, POR FAVOR";N$: DIM A(10): D
IM J$(10,4): DIM R(10,10): IF N$
="CLAVE" THEN LET CV=1
5260 IF N$="CLAVE TOTAL" THEN LE
T CV=2
5270 [illegible] THEN GOTO 7
000
5290 RETURN
5300 FOR Z=1 TO LEN A$: SOUND .0
01,2: PRINT A$(Z);: SOUND .025,2
0: NEXT Z: PRINT : PRINT : RETUR
N
6000 DIM T$(10,32): DIM I$(10,22
4): DIM P$(10,10,13): DIM O$(10,
10,13): DIM C(10,10): FOR X=1 TO
 10
6010 CLS : IF CV=1 THEN GOSUB 64
00
6011 IF CV=1 THEN LET Z$="PULSE
EL NUMERO A MODIFICAR": GOSUB 99
00: PAUSE 0: LET E$=INKEY$: IF C
ODE E$<48 OR CODE E$>57 THEN GOT
O 6011
6012 IF CV=1 THEN LET X=VAL E$:
LET X=X+1
6013 CLS : INPUT "TITULO TEMA ";
 INVERSE 1;(X-1); INVERSE 0;" ",
T$(X)'"INSTRUCCIONES -MAX 7 LINE
AS-",I$(X)
6015 FOR Y=1 TO 10
6020 INPUT "PREGUNTA "; INVERSE
1;(CHR$ (Y+64)); INVERSE 0,P$(X,
Y)'"OPCION "; INVERSE 1;(Y-1); I
NVERSE 0,O$(X,Y)'"OK? N=NO",K$:
IF K$="N" THEN GOTO 6020
6030 NEXT Y: GOSUB 6600: IF CV=2
 THEN NEXT X: RETURN
6040 PRINT AT 21,3;"OTRA MODIFIC
ACION? S=SI": PAUSE 0: IF INKEY$
<>"S" THEN RETURN
6050 GOSUB 6010: RETURN
6210 CLS : FOR X=1 TO 10: PRINT
AT 2+X-1,0;J$(X);TAB 5; INVERSE
1;CHR$ (X+64); INVERSE 0;" ";T$(
X): FOR Z=20 TO 40 STEP 2: SOUND
 .005,Z: NEXT Z: NEXT X
6220 PAUSE 15: IF INS=0 THEN LET
 Z$=N$+", ELIJE EL TEMA SOBRE EL
 CUAL DESEAS CONTESTAR, PULSANDO
 LA TECLA QUE CORRESPONDA. ADELA
NTE ...": GOSUB 9985
6225 IF INS=1 THEN LET Z$=N$+",
EL SIGNO '+' INDICA LOS TEMAS QU
E YA HAS CONTESTADO, Y EL PUNTAJ
E QUE HAS OBTENIDO.": GOSUB 9985
: LET INS=2
6230 PAUSE 0: LET Z$=INKEY$: IF
CODE Z$<65 OR CODE Z$>74 THEN GO
TO 6230
6240 LET X=CODE Z$-64: LET J$(X,
1)="+": FOR Z=1 TO 10: SOUND .02
,Z: NEXT Z: RETURN
6400 FOR X=0 TO 9: PRINT AT X+2,
5; INVERSE 1;X; INVERSE 0;" ";T$
(X+1): GOSUB 6950: NEXT X: RETUR
N
6500 CLS : LET Z$=T$(X): GOSUB 9
998: GOSUB 9900: FOR Y=1 TO 10:
GOSUB 6950: PRINT INVERSE 1;CHR$
 (Y+64); INVERSE 0;" ";P$(X,Y);T
AB 17; INVERSE 1;Y-1; INVERSE 0;
" ";O$(X,Y)'': NEXT Y: RETURN
6600 CLS : GOSUB 6500: FOR Y=1 T
O 10: INPUT "RESPUESTA CORRECTA
DE "; INVERSE 1;(CHR$ (Y+64)); I
NVERSE 0;" EN ";C(X,Y): LET C(X,
Y)=C(X,Y)+1: NEXT Y: RETURN
6800 REM ## FIN INTERVENCION ##
6810 CLS : LET Z$="RESPUESTAS DE
 "+N$: GOSUB 9900: PRINT
6820 LET TP=0: LET TA=0: FOR X=1
 TO 10: IF R(X,1)=0 THEN GOTO 68
40
6830 LET Z$=T$(X): GOSUB 9998: F
OR Z=40 TO 44: SOUND .05,Z: NEXT
 Z: PRINT Z$;TAB 28-LEN STR$ A(X
);A(X);" PS": LET TP=TP+1: LET T
A=TA+A(X)
6840 NEXT X: LET PR=TA/TP: LET P
R=INT (PR*100+.5): LET PR=PR/100
6850 PAUSE 30: PRINT : PRINT : G
OSUB 6950: PRINT "CONTESTO: ";TP
;: IF TP=1 THEN PRINT " VEZ"
6855 IF TP>1 THEN PRINT " VECES"
6860 PAUSE 30: PRINT : GOSUB 695
0: PRINT "OBT[illegible]";TA;: IF TA=1
 THEN PRINT " PTO"
6865 IF TA<>1 THEN PRINT " PTOS"
6870 PAUSE 30: PRINT : GOSUB 695
0: PRINT PAPER 1;"PROMEDIO GENER
AL: ";PR;: IF PR=1 THEN PRINT PA
PER 1;" PUNTO"
6873 IF PR<>1 THEN PRINT PAPER 1
;" PUNTOS"
6880 IF PR<=3 THEN LET Z$=N$+",
TU PROMEDIO ES 'MALO'. NO IMPORT
A, DEBES INTENTARLO OTRA VEZ. AD
ELANTE ...": GOSUB 9985
6882 IF PR>3 AND PR<=5 THEN LET
Z$=N$+", TU PROMEDIO ES 'REGULAR
'. RECUERDA QUE PUEDES MEJORARLO
 EN OTRO INTENTO. GRACIAS POR PA
RTICIPAR ...": GOSUB 9985
6884 IF PR>5 AND PR<=6 THEN LET
Z$=N$+", TU PROMEDIO ES 'BUENO'.
 SEGURAMENTE LO MEJORARAS LA PRO
XIMA VEZ. GRACIAS POR PARTICIPAR
 ...": GOSUB 9985
6886 IF PR>6 AND PR<=7 THEN LET
Z$=N$+", TU PROMEDIO ES 'MUY BUE
NO'. TE FELICITO POR LO MUCHO QU
E SABES.": GOSUB 9985
6888 IF PR>8 THEN LET Z$=N$+", T
E FELICITO!!!. TU PROMEDIO ES 'E
XCELENTE'. TIENES MUY BUENOS CON
OCIMIENTOS SOBRE ESTOS TEMAS. SI
GUE ASI ...": GOSUB 9985
6890 GOSUB 6950: PRINT AT 21,2;"
CONTESTA OTRA PERSONA? S=SI": PA
USE 0: IF INKEY$="S" THEN GOTO 2
5
6900 GOTO 1
6950 FOR Z=20 TO 30 STEP 3: SOUN
D .04,Z: NEXT Z: RETURN
7000 INPUT "NOMBRE DE ARCHIVO -M
AX 10 LETRAS";E$
7010 PRINT AT 10,7;"Desconecte
EAR'",AT 13,3;"para archivar ";
FLASH 1;">"; FLASH 0;E$; FLASH 1
;"<"
7020 SAVE E$ LINE 1
7030 CLS : PRINT AT 21,8;"VERIFI
CA? N=NO": PAUSE 0: IF INKEY$="N
" THEN GOTO 1
7040 CLS : PRINT AT 10,5;"Recone
cte 'EAR'";AT 13,2;"para verific
ar "; FLASH 1;">"; FLASH 0;E$; F
LASH 1;"<"
7050 VERIFY E$: GOTO 1
8000 POKE USR "N"+0,30: POKE USR
 "N"+1,64: LET Z$="EDUCAR": PRIN
T AT 21,0; INK 0;Z$
8010 LET T=30: LET Z1=LEN Z$: FO
R W=0 TO 8*Z1-1: FOR Q=0 TO 7
8015 LET TIN=INT (RND*5)+2
8020 IF POINT (W,Q)=0 THEN GOTO
8060
8030 INK TIN: PLOT W*4+28,Q*4+14
0-T: DRAW 4,0: DRAW 0,4: DRAW -4
,0: DRAW 0,-3: DRAW 3,0: DRAW 0,
2: DRAW -2,0: DRAW 0,-1: DRAW 2,
0: DRAW -2,-2
8040 DRAW 5,5: DRAW 0,4: DRAW 0,
-4: DRAW 4,0: DRAW 0,4: DRAW 0,-
4: DRAW -5,-5
8050 DRAW 0,4: DRAW 5,5: DRAW -4
,0: DRAW -5,-5
8055 SOUND .005,45
8060 NEXT Q: NEXT W: PRINT AT 11
,0;"";
8070 RETURN
9900 DIM W$(32): PRINT INVERSE 1
;(W$( TO 16-(LEN Z$/2))+Z$+W$( T
O 16-(LEN Z$/2)))( TO 32); INVER
SE 0: RETURN
9985 LET P=LEN Z$: LET CO=1
9986 IF CO<31 THEN PRINT AT 21,(
31-CO); INK TI;Z$(1 TO CO)
9987 IF CO>=31 AND CO<=LEN Z$ TH
EN PRINT INK TI;AT 21,0;;Z$(CO-3
0 TO CO)
9988 IF CO>=LEN Z$ THEN LET Z$=Z
$+" "
9989 IF LEN Z$=P+32 THEN RETURN
9990 SOUND .05,35: LET CO=CO+1:
GOTO 9986
9998 FOR Z=LEN Z$ TO 1 STEP -1:
IF Z$(Z)<>" " THEN RETURN
9999 LET Z$=Z$( TO Z-1): NEXT Z
```

K64

En esta sección atendemos todas aquellas consultas y sugerencias que nuestros lectores deseen realizar. Para ello sólo debe dirigirse a esta redacción, sección "Consultas".

SAVE

En primer lugar deseo felicitarlos por su revista y pedirles que sigan publicando programas para Sinclair tanto 1000 como 1500.

Una pregunta sobre la introducción en cassettes: ¿Se puede tipear primero lo que se desea introducir y una vez terminado, recién introducirlo en el cassette? Esto no lo tengo muy en claro ya que el manual no es muy explícito para principiantes.

¿Dónde puedo conseguir el Nº 1?

Horacio Mercado
ROSARIO. SANTA FE

K-64

La computadora grabará en cassette usando la función SAVE, aquello que tenga en su memoria en ese momento, así lo haya tecleado o lo haya cargado antes de otro cassette. LAMENTABLEMENTE EL Nº 1 ESTA AGOTADO.

CIRCUITO

He armado el sencillo circuito que ustedes publicaron en este número, pero no he obtenido los resultados necesarios; en una palabra no funciona. Hice un montón de pruebas dentro de mis conocimientos de electrónica, pero sigue negativo. A lo mejor hay algún valor de capacitor que no coincide. El circuito es el de DECK (omití mencionarlo).

K-64

El circuito al que se refiere, funciona bien y no hay omisiones o errores de impresión.

Sin embargo, aquí van unos consejos para que pruebe:

* La tensión de alimentación debe ser de 5V, estabilizados.
* La señal de entrada debe tener por lo menos 0,5 Vpp.
* Deben unirse las salidas de ambos canales del deck.
* Puede intentar aumentar la ganancia o sensibilidad del circuito, cambiando R3 A 1800 OHM.
* Verifique que el resto de los componentes esté en buen estado, y conectados correctamente según su polaridad.

Esperamos que con estos datos pueda resolver el problema que comenta. Hasta pronto.

ADOLFO L. de ARRIBA
Pergamino - Bs. As.

CARACTERES GRAFICOS

En el número 3, en el programa AM Spriter y Tragamonedas, hay una serie de líneas en las que hay caracteres gráficos especiales de Commodore que, supongo, por una cuestión de impresión, no resultan legibles y en el caso de Tragamonedas no se ve el efecto del programa. ¿Qué debo hacer?

Alberto Martínez
Florida

K 64

Ese es un problema de compatibilidad de los caracteres gráficos especiales de Commodore y la impresora que utilizan Carlos Ay y Daniel Manduca para listar los programas que presentan en la revista.

Con respecto al AM Spriter, los caracteres gráficos que no se ven son intrascendentes y no afectan al desarrollo y ejecución del programa.

En relación con Tragamonedas, publicamos en esta página una modificación del listado del programa.

Ingresando las líneas del listado 1 podrán tener el juego totalmente completo y empezar a derrochar los viejos $a allí. Buena suerte.

LISTADO NUMERO 1

```
9 FR$(1)="{RED}{RVON} "+CHR$(193)+" ":FR
$(2)="{CYAN}{RVON} "+CHR$(218)+" "
10 FR$(3)="{YELO}{RVON} "+CHR$(216)+" ":
FR$(4)="{ORNG}{RVON} "+CHR$(211)+" "
11 AR$=CHR$(207)+CHR$(183)+CHR$(183)+CHR
$(183)+CHR$(208)+"     "
12 ME$=CHR$(180)+"    "+CHR$(170)+"     "
13 AB$=CHR$(204)+CHR$(175)+CHR$(175)+CHR
$(175)+CHR$(186)+"     "
22 FORI=1TO4:FE$(I)=FR$(I):NEXTI
1040 F=9:C=0:GOSUB100:FORI=0TO39:PRINTCH
R$(184);:NEXTI
1050 F=11:C=3:GOSUB100:PRINTCHR$(207);
1060 FORI=1TO19:PRINTCHR$(183);:NEXTI
1070 PRINTCHR$(208)
1080 F=12:GOSUB100:PRINTCHR$(180);"SU SA
LDO: $A          ";CHR$(170)
1090 F=13:C=3:GOSUB100:PRINTCHR$(204);
1100 FORI=1TO19:PRINTCHR$(175);:NEXTI
1110 PRINTCHR$(186)
1120 F=15:C=3:GOSUB100:PRINTCHR$(207);
1130 FORI=1TO31:PRINTCHR$(183);:NEXTI
1140 PRINTCHR$(208)
1150 F=16:GOSUB100:PRINTCHR$(180);"
                              ";CHR$(170)
1160 F=17:C=3:GOSUB100:PRINTCHR$(204);
1170 FORI=1TO31:PRINTCHR$(175);:NEXTI
1180 PRINTCHR$(186)
1190 F=19:C=3:GOSUB100:PRINTCHR$(207);
1200 FORI=1TO20:PRINTCHR$(183);:NEXTI
1210 PRINTCHR$(208)
1220 F=20:GOSUB100:PRINTCHR$(180);"SU GA
NANCIA: $A      ";CHR$(170)
1230 F=21:GOSUB100:PRINTCHR$(204);
1240 FORI=1TO20:PRINTCHR$(175);:NEXTI
1250 PRINTCHR$(186):S=1000:G=0
```




INTERCAMBIO DE PROGRAMAS

Primero que todo quisiera disculparme por creer que la tardanza de los ejemplares era de su parte.
Las revistas son magníficas, no cabe duda de que su esfuerzo por lanzarlas al mercado no ha sido en vano, debido a que son un verdadero éxito.
Junto con esta carta les envío el programa para el concurso. Este programa no es nada extraordinario porque soy un "principiante", pero mi deseo no es ganar sino participar para integrarme al mundo de la computación. Lo hice en una computadora TI-99/4A del instituto al cual concurro, ya que no poseo una.
También les mando el cupón del sorteo mensual (espero que acepten una fotocopia de él, porque no deseo recortar la página).
Desearía que publicaran que quiero intercambiar programas varios, mi nombre y dirección:

GONZALO VILLAFAÑE
Alberdi 457
San Francisco
(2400) CORDOBA

K-64

Gracias Gonzalo por tu aliento y tu colaboración. Con respecto a los cupones, no hay problema en enviar fotocopias.

ASSEMBLER

Quiero decirles que su revista es muy interesante, ya que hace rato que estoy buscando alguna publicación de este tipo y ésta, por lo que ví hasta ahora, es la que más me gustó. El artículo "Conociendo las Computadoras" está muy bueno. Quiero felicitarlos también por el programa "Práctica de paracaidismo", que funciona muy bien.

Quisiera que publiquen o me envíen si les es posible información sobre cómo se usa el lenguaje Assembler y cuáles son sus ventajas y desventajas, y cómo se implementa y usa en un computador TI-99/4A.

Darío Tamagnini
San Francisco - Córdoba

K-64

El uso de Assembler en cualquier computadora, permite la explotación al máximo de su capacidad operativa.

Sin embargo se requieren profundos conocimientos sobre el tema y sobre el funcionamiento interno de la computadora, así como también la ayuda de programas especiales.

Lamentablemente para la TI99/4A no se dispone de facilidades para ello, debido a que se trata de una máquina que se dejó de fabricar en Estados Unidos y por lo tanto no se recibe más ayuda ni software adecuado para profundizar en este lenguaje de programación.

INTERFERENCIA

Los felicito por vuestra revista y a la vez consulto si conocen que se comercialice un teclado tipo profesional para la TS 1000, o que sea posible armarlo con teclas independientes. También creo que sería bien aceptado un artículo sobre la forma de solucionar la interferencia de R.F. que producen en la pantalla del televisor (no inestabilidad vertical) las TS 1000 y TS 1500?

K-64

No sabemos de nadie que comercialice teclados de ese tipo.
Sí se puede sin embargo, adaptar teclados de reza antigua, o armarlo a partir de teclas individuales de máquina de calcular. Pero es de esperar que al hacer las cuentas del material y tiempo insumido en fabricarlo, no se justifique realmente.
El problema de la interferencia que sale en la pantalla, es bastante complejo de resolver, pero aquí van algunos consejos:

- Cambiar el cable coaxil por otro más largo (puede usarse un buen cable blindado tipo micrófono), e ir acortando su longitud hasta lograr mejor imagen.
- Enrollar el cable en una barrita de ferrite (algunas vueltas)
- Idem en un anillo de ferrite.
- Usar otro adaptador de coaxil/30052, o directamente probar sin él.
- Tratar de llevar la salida del modulador a otro canal, desatornillando el núcleo de la bobina del mismo.
- Hay televisores que se ven fatal (algunos NOBLEX, por ejemplo)
- Soldar dentro del modulador, en el conector de salida, un capacitor de bajo valor (probar con 47PF)
- Obviar el modulador y parte del TV, conectándolos directamente por video.

¡Esperamos que alguna le sirva!

TS2068: LA INTERFASE 1

Me gustaría saber si la interface "ZX 1" para la conexión de "Midrodrives" de la computadora Spectrum es compatible con la computadora TS 2068. Además tengo la duda de si un modem se conecta directamente en las conexiones de la computadora Spectrum o hace falta alguna interface. Una vez conectado la Spectrum puede comunicarse con una Texas TI99/4A; con una Apple o con una IBM, para acceder a información intercambio de programas, etc.

Néstor Hugo López Cabanillas
LOMAS DE ZAMORA
Buenos Aires

K-64

La interface 1 no es compatible con la TS 2068. Además, por lo que sabemos no estará disponible a la venta por algunos meses. Por otro lado, disponiendo de una ZX Spectrum con la Interface 1, se le puede conectar cualquier modem que tenga entrada RS-232 y su Software adecuado para controlarlo. También existen Modems que no necesitan interfaces y se conectan directamente en el conector posterior. En cualquiera de los casos, es posible conectarse con cualquier otra computadora que tenga conectado un Modem que trabaje en la misma norma, (Bell 103 ó CCITT), como así también comunicarse con Bases de datos internacionales.

PROGRAMA PARA TELEGRAFIA

De algunas cartas enviadas a esta redacción por lectores que han tenido dificultades con el programa para telegrafía para computadores TS1000, creemos necesario tener en cuenta los siguientes factores:

a) Para los que no utilizan un programa Ensamblador, en el ingreso del programa, es decir que "Pokean" los códigos del lenguaje de máquina directamente, tener en cuenta que, antes de ingresar los códigos se debe generar una instrucción 1 REM con tantos espacios como códigos se vayan a tipear, pues caso contrario el intérprete BASIC se confundiría y se perdería el control de la máquina, debiéndosela apagar para retomarlo.

b) El programa ha sido chequeado en la práctica durante bastante tiempo y aunque suene redundante FUNCIONA BIEN, siempre que se tengan en cuenta su rango de acción y limitaciones, es decir:

1) La señal recibida debe ser MUY FUERTE y estar por encima del ruido de fondo en por lo menos 30 dB.
2) La transmisión debe ser razonablemente bien hecha en cuanto a calidad de manipulación.
3) El nivel de audio proveniente del receptor debe



Armstrong: "Trabajo visionario".

NUEVOS DISPOSITIVOS

La investigación en Yorktown está dividida principalmente en cuatro áreas: tecnología avanzada en chips; tecnología del encapsulado; materiales y procesos, y ciencia del semiconductor.

Tecnología avanzada en chips es lo que su nombre indica: la invención y desarrollo de nuevos dispositivos semiconductores y procesos que eventualmente puedan ser usados en computadoras IBM. Los chips resultantes deberán ser juntados en un módulo y los módulos en una plaqueta, que se hallará interconectada, a su vez, con otras y con el mundo exterior.

La tarea del encapsulado, por su parte, es la que permite hallar los medios de "empacar" los circuitos lo más densamente posible, de modo de hacer a una unidad simple y rápida a la vez.

Los materiales elegidos para desarrollar nuevas tecnologías y nuevos encapsulados, tienen un enorme impacto en el funcionamiento del producto final, como así también el proceso usado en la producción. "Si se usa el material adecuado y el proceso correcto, la Naturaleza estará de nuestro lado", dice Joseph Logue, asistente de director de APTL de IBM. "Si elegimos equivocadamente, tendremos problemas..."

MAXIMA POTENCIA, MINIMO ESPACIO

Recientemente, un grupo dedicado al estudio avanzado en tecnología del semiconductor dio un gran salto en esa dirección, en un proyecto que forma parte del programa de gobierno "VHSIC" (very high speed integrated circuits).

Ellos produjeron circuitos en los cuales la menor dimensión es 0.5 micrón. Estos circuitos están hechos a base de tecnología del tipo n-MOS FET. Aunque ya estaba desarrollada esta tecnología anteriormente, nadie aún había logrado estos niveles de integración y a tan bajo consumo de corriente.



Tareas de inspección en la fabricación de chips del tipo VLSI.

Otra tecnología que aparece en competencia es la del tipo CMOS, que resulta en dispositivos más complejos pero de menor consumo. Este mismo grupo también está aplicando su experiencia de n-MOS en el área de los CMOS, para desarrollar dispositivos y memorias de alta velocidad para computadoras.

UN CANDIDATO POTENCIAL: ARSENIURO DE GALIO

A pesar que ahora domina la tecnología basada sólo en el silicio, parece que se ha llegado al techo en lo que se refiere a velocidad y performance con este material. Los investigadores están ahora entusiasmados con otras aleaciones de semiconductores. Entre ellas el Arseniuro de Galio, es objeto de intensos estudios.

El arseniuro de galio ofrece intrínsecamente mayor velocidad que el silicio. Simplemente, los electrones se mueven más rápidamente en él. Se han logrado al día de hoy con tecnología MESFET (metal semiconductor FET), tiempos de cinmutación menores a 20 picosegundos! (un picosegundo es 1/1.000.000.000.000 de segundo...) Estos dispositivos además consumen menos corriente que sus equivalentes de silicio. De ahí también su atractivo.

El silicio ha venido siendo estudia-

SORTEO - ENCUESTA K64

LLENE ESTE CUPON Y PARTICIPE DEL SORTEO MENSUAL

PREMIOS:

20 CASSETTES Y 10 BECAS PARA CURSOS BASIC

NOMBRE .... Edad ....

Domicilio .... TE.: ....

C.P. .... Localidad ....

Enviarlo a:
K64 Computación Para Todos
Cerrito 1320 1° (1010) Buenos Aires REP. ARGENTINA

ENCUESTA

COMPUTADORA: ☐ CZ 1000 ☒ CZ 1500 ☐ CZ 2000 ☐ TIMEX 2068 ☐ TK 83
☐ TK 85 ☐ TK 90 ☐ C-16 ☐ C-64 ☐ TI 99/4A
☐ Otras .... ☐ NO TENGO AUN

ME GUSTARIA VER:

MAS	IGUAL	MENOS	
☐	☐	☒	PROGRAMAS DE APLICACION ESPECIFICA
☒	☐	☐	PROGRAMAS EN BASIC
☐	☐	☒	PROGRAMAS EN LOGO
☐	☒	☐	PROGRAMAS EN LENGUAJE DE MAQUINA
☒	☐	☒	PROGRAMAS EN OTROS LENGUAJES
☐	☐	☐	ANALISIS DETALLADOS DE LOS PROGRAMAS
☐	☒	☐	NOTAS PARA BEGGINERS
MAS	IGUAL	MENOS	
☒	☐	☐	JUEGOS
			CALIFICACION DESCRIPTIVA DE:
☒	☐	☐	PROGRAMAS DE JUEGO
☐	☐	☒	PROGRAMAS DE APLICACIONES COMERCIALES
☐	☐	☒	PROGRAMAS EDUCATIVOS
☒	☐	☐	HARDWARE

QUE ES LO QUE MAS TE GUSTA DE K64?

QUE ES LO QUE MENOS TE GUSTA?

ser muy alto requiriéndose entre 15 y 20 VPP para el óptimo funcionamiento.

4) El tono de la señal debe ser agudo, mucho más del que se utiliza para tomar telegrafía a "oído", algunos receptores tendrán inconvenientes para proporcionar este tono cumpliendo 1) y 3).

c) El nivel que se obtiene como salida en transmisión del conector MIC es exiguo (algunas decenas de milivoltios) y no podrá en la mayoría de los casos excitar adecuadamente el micrófono de un tranceptor de BLU, debiéndose para ello utilizar un pequeño amplificador de audio.

d) La utilización del programa con un tranceptor de BLU no genera señales de tipo R2 (ilegales) sino RI que son perfectamente legales, no obstante es conveniente cuando el uso del programa exceda la corta experimentación el proceder al control de la manipulación mediante un rectificador de audio y la clavija KEY del emisor.

e) El programa ayuda a la introducción en la materia, pero no implica que para un trabajo a largo plazo no se deba aprender Morse por los métodos tradicionales, en tal eventualidad el mismo es un excelente auxiliar.

f) No en todos los receptores de TV se obtiene una nota limpia por el parlante de audio y a modo de monitor, en algunos esto ocurre sin inconvenientes, en otros es necesario regular el control de sintonía y para que la nota emitida sea limpia la imagen de video es mala, y en otros casos (ciertos televisores color) independientemente de lo que se haga no hay forma de lograr sonido alguno.

COMPATIBILIDAD

Tengo una computadora microdigital TK 2000 y quiero preguntarles qué software de otras marcas son compatibles con la mía; ya sean juegos o de otras aplicaciones.

Diego Verruno
Témperley - Pcia. Bs. As.

K-64

Diego, la TK 2000 no tiene mucha compatibilidad con otras marcas. Sólo podrían funcionar algunos cassettes de computador Apple.



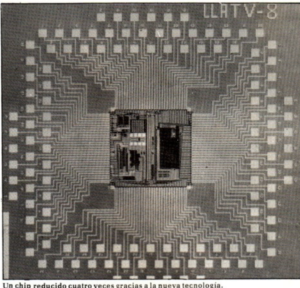
do hace 20 años; el arseniuro de galio en cambio recién está en su infancia.

"Básicamente, debemos reinventar la rueda", dice Dean Eastman, director del APTL y jefe del programa sobre Arseniuro de Galio.

Seguramente estos dispositivos estarán funcionando pronto en las futuras computadoras, sumergidos en nitrógeno líquido, que es donde se logran las mayores velocidades de trabajo.

Pero el trabajo no es fácil, aún existen muchas barreras tecnológicas que saltar. Todos sabemos mucho sobre el silicio, pero poco sabemos respecto a esta nueva posibilidad.

También se hallan aplicaciones en áreas tales como microondas y la industria de la opto-electrónica. Mientras tanto, los científicos continúan enfatizando ambas tecnologías; exprimiendo aún más las posibilidades del silicio, y avizorando las del arseniuro de galio.



Un chip reducido cuatro veces gracias a la nueva tecnología.

GLOSARIO

LETRA "E"

EDIT:

Corrección de programas o parte de ellos.

EDITOR PROGRAM:

Programa que permite la manipulación de texto para su corrección mientras aún se encuentra en la memoria de la máquina.

EMULATOR:

Dispositivo o programa que hace que un ordenador simule el comportamiento de otro.

EPROM: ERASABLE PROGRAMMABLE READ ONLY MEMORY:

Una memoria que puede ser grabada y usada como una ROM y que puede volverse a programar, borrándola previamente con luz ultravioleta.

ERGONOMIC:

Diseño de equipos o accesorios de uso intensivo, de modo que resulten de uso cómodo y descansado, reduciendo así el esfuerzo físico, visual e intelectual exigido al operador.

EVEN PARITY:

Condición que ocurre cuando la suma de bits puestos a uno en un byte, es par (el cero se considera par).

EXCLUSIVE OR:

Operación lógica de Bool. Resulta un uno lógico cuando en cualquiera de los registros de entrada haya un uno (en los dos a la vez no). En cualquier otra situación el resultado es cero.

EXECUTIVE PROGRAM:

Programa o parte de programa que se encarga de controlar la operación general de un programa o sistema complejo.

FILE GAP:

Pequeños trozos de cinta sin grabar usados para separar archivos o blocks de datos en un medio magnético.

FILES:

Se refiere esencialmente a blocks de información formando un "record" o grupo de datos. Archivo.

FIRMWARE:

Un programa suministrado dentro de una ROM. Referido a software no modificable, grabado en forma permanente.

FLAGS:

Celdas de almacenamiento de un solo BIT, que sirven para indicar si existen o no ciertas condiciones lógicas luego de realizadas operaciones en la CPU.



FROGGER

Comp: CZ1000/1500 TK83/85
Conf: 2 K
Clas: Entretenimiento

INSTRUCCIONES:

Esta es una de las tantas versiones del conocido "programa de la **ranita**".
Hay que guiar la "RANA" (F) para que cruce una calle sin ser atropellada. Los comandos son las teclas 5 y 8 para ir a la izquierda y derecha, y la tecla 7 hace avanzar la rana.

VARIABLES:

A$,B$: representan los vehículos de la calle.
A,B: mantienen las coordenadas de posición de la rana.
C$: memoriza el número de la tecla pulsada.
S: mantiene el score.

PROGRAMA:

Líneas: 10 a 60: Generan la pantalla en posición de comienzo.
INKEY$ es el comando que lee el teclado e indica qué tecla es pulsada.

70: Muestra en la pantalla los vehículos.

80: Esta línea lee (PEEK) en la parte de la memoria que corresponde a la pantalla la próxima posición de PRINT AT. Si el valor es mayor, en ese lugar de la memoria que 127, significará que el auto se superpone a la rana, y el programa parará.

90: Borra la última posición de la rana.

100: Bifurca a la rutina de avance de la rana si se ha pulsado la tecla "7".

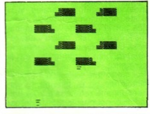
110: Genera las nuevas coordenadas para desplazar la rana hacia los costados, en caso de haber pulsado las teclas "5" ó "8".

120: Imprime la rana en la nueva posición.

130 a 150: Generan la nueva pantalla y regresa a la línea 60 para recomenzar la rutina.

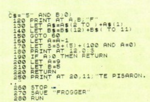
160 a 220: Incrementa "S" si la rana se movió hacia adelante e imprime el score.

PANTALLA





```
  1 REM    ****  K 64  ****
  2 REM    ***  FROGGER ***
  3 REM    ****************
 10 LET A$="▄▄▄  ▄▄▄  "
 20 LET B$="  ▄▄▄  ▄▄▄"
 30 LET S=0
 40 LET A=9
 50 LET B=6
 60 LET C$=INKEY$
 70 PRINT AT 1,0;A$,,,B$,,,A$
,,,B$;AT A,B;
 80 IF PEEK (PEEK 16398+256*PEE
K 16399)>127 THEN GOTO 250
 90 PRINT AT A,B;" "
100 IF C$="7" THEN GOSUB 160
110 LET B=B+(C$="8" AND B<11)-(
```



```
C$="5" AND B>0)
120 PRINT AT A,B;"F"
130 LET A$=A$(2 TO )+A$(1)
140 LET B$=B$(12)+B$( TO 11)
150 GOTO 60
160 LET A=A-1
170 LET S=S+(5)+(100 AND A=0)
180 PRINT AT 12,0;S
190 IF A>0 THEN RETURN
200 LET A=9
210 LET B=5
220 RETURN
250 PRINT AT 20,11;"TE PISARON.
."
260 STOP
270 SAVE "FROGGER"
280 RUN
```

EXAMEN

COMP.: TS 2068
CONF.: 48 K
CLAS.: EDU

Alumnos del curso de Derecho Procesal II de la Facultad de Ciencias Jurídicas de la Universidad del Salvador, fueron evaluados mediante el uso de una computadora.
La tarea estuvo a cargo del Dr. Luis M. Gaibrois — FOTO —, profesor titular de la materia, quien utilizó una microcomputadora hogareña de su propiedad (TS 2068) y un programa en lenguaje BASIC que él mismo diseñara tras varios meses de labor (lo publicamos por separado).
El programa estructura las preguntas que conforman el examen (en esta ocasión fueron 25) de forma tal que aparecen en pantalla numeradas, y conjuntamente con la alternativa de dos, tres o más respuestas, de las cuales tan sólo una es la correcta.
También permanece en pantalla a lo largo de la examinación, un señalador de la cantidad de respuestas correctas e incorrectas. Finalizado el interrogatorio, aparece un "certificado" con el nombre del alumno, y en el que consta el resultado final. Luego, ingresada la calificación, se extiende la constancia respectiva mediante el uso de una impresora térmica (Alfacom 32).
Terminado el turno de exámenes, luego de rendir el último alumno, aparece en pantalla el listado de examinados y las notas respectivas, lo cual equivale, una vez copiado por la impresora, el "acta volante" que confecciona habitualmente todo tribunal examinador.
Y si bien el sistema de preguntas con "respuestas por alternativa" se ha venido utilizando desde tiempo atrás en exámenes escritos, su implementación a través de una computadora le otorga características propias, que lo diferencian por completo de los dos sistemas tradicionales (el oral y el escrito).
A la imparcialidad del examen escrito, se le agrega en este caso la garantía de la "infalibilidad", desde que las preguntas han debido ser necesariamente preparadas con anticipación más que suficiente como para verificar su acierto académico.
No se dá tampoco en la máquina el



PANTALLA

```
(Para comenzar: pulse 'ENTER').
 FACULTAD DE CIENCIAS JURIDICAS

 El alumno hugo smash
 ha rendido en la fecha examen
 parcial de la asignatura Dere-
 cho Procesal Civil y Com., ha-
 biendo respondido  4  veces en
 forma correcta, y  1  veces en
 forma incorrecta.

5/marzo/1985

                Dr.Luis M.Gaibrois

(Sr.Profesor: ingrese la cali-
ficacion del alumno y ENTER).
```

cansancio del examen oral, que de prolongarse, se torna inevitablemente altamente discrecional.
Por otra parte existe una exigencia intelectual muy severa para el alumno, muy difícil —por no decir imposible— de obtener mediante la evaluación escrita, como es la necesidad de responder a una orden secuencial inalterable. Es decir las preguntas se suceden una tras otra y no pueden "saltearse" como es costumbre actual en los escritos.
Pero, como contrapartida, juega a favor del alumno la posibilidad de regular su tiempo de respuesta, por cuanto el manejo del teclado le pertenece por completo reservándose claro está el profesor la tecla 0 para dar por terminado en cualquier momento el examen con el clásico "suficiente" si lo considera apropiado.
No existen las distracciones propias del examen oral, ni las nefastas influencias visuales.
Pero además de otros aspectos docentes, una particularidad en el examen por computadora garantiza su vigencia en lo futuro: es la obtención inmediata del resultado (situación ésta que no se puede dar en el examen escrito, y muy difícilmente se logrará en un examen oral).
Esta rapidez es la más notoria de sus ventajas, la cual a su vez produce dos fenómenos pedagógicos de indudable repercusión.
Uno se da cuando el marcador de respuestas incorrectas ha ido creciendo a lo largo del examen, de forma tal que es el mismo estudiante quien va asimilando paulatinamente el aplazo, sin tener que sufrir la pequeña humillación, de la más que común inflexión de reproche que se acostumbra formular en el examen oral. O la tediosa espera, a veces de días, cuando no semanas, del resultado de la correccción del examen escrito, que cuando llega desvincula el esfuerzo del resultado, y prácticamente carece de interés.
El otro fenómeno ocurre cuando el marcador de las respuestas correctas le es favorable al alumno. Entonces se advierte que "constituye un formidable acicate para seguir o profundizar, ánimo vital éste, que tan pocos profesores insuflan, tan retaceado por otros, como inexistente en los demás", según Gaibrois. "Y la gratificación entonces, nace plena y espontánea en el alumno, que por otra parte la merece, cara a cara con su profesor, porque el estudio, como el trabajo, es el fruto de un esfuerzo enaltecedor", agregó.
La máquina programada con preguntas, excluye la improvisación en la formulación de las mismas, característica que presentan muchos interrogatorios orales, y si existe repetición ésta no es conocida por el resto de los examinados (que de existir, obliga a una tediosa búsqueda de novedades, no siempre bien encontradas en el momento).
"Por cierto que existen grandes limitaciones en el sistema, fundamentalmente referidas a la examinación conceptual o teórica del estudiante. Por ello se presta muy bien al examen parcial de la materia como en este caso, constituyendo un auxiliar inestimable de la docencia", reconoción el profesor.
"La máquina atrae al estudiante —advirtió—. Y en este caso se ha advertido en la juventud, una especie de renacer del coraje criollo, un deseo de vencer el miedo a lo desconocido y porque no decirlo, un beneficioso sistema de competir, de 'ganarle a la máquina'."
"Si el deseo de ganar que se genera, se opera a través del conocimiento, no puede haber máquina superior. Se habrá programado para mejorar", concluyó.





```
  1 REM examen
  3 PRINT "▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚": PRINT AT 5,6;"PRO
GRAMA EDUCATIVO": PRINT AT 10,5;
"by Luis M. Gaibrois.": PRINT AT
 15,6;"(Abril-Mayo/1985)."
  4 PRINT AT 20,0;"▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚": PRINT AT
21,0;"  (Para seguir pulse 'ENTE
R')."
  5 PAUSE 0
  6 CLS
 11 PRINT AT 0,1;"1) Tipee EN N
UMEROS,la cantidad ": PRINT AT 1
,3;" de alumnos que seran someti
-": PRINT AT 2,3;" dos a esta ev
aluacion:"
 12 INPUT X: PRINT AT 4,10;X
 13 PRINT AT 7,1;"2) Tipee la f
echa del dia de       hoy (Por e
j.:5/Junio/1985).": PRINT AT 11,
0;"
"
 14 INPUT t$: PRINT AT 10,10;t$
: PRINT AT 15,0;"
"
 15 PRINT AT 17,0;"(Sr.Profesor
: recuerde que comenzadas las pr
eguntas,en cualquiermomento Ud.
puede dar por termi-nado el exam
en pulsando '0')."
 16 PRINT AT 12,0;"Verifique.Si
 se ha equivocado altipear:pulse
 'si',ENTER,y vuelvaa escribir.(
Seguir: pulse ENTER)"
 17 INPUT e$
 18 IF e$="si" THEN GO TO 11
 20 DIM N$(X,20)
 25 DIM C(X)
 30 LET A=0
 35 LET A=A+1
 48 CLS
 49 POKE 23609,100
 50 REM =Logotipo presentacion.
 55 PRINT "
"
 60 PRINT AT 1,1;"FACULTAD DE C
IENCIAS JURIDICAS"
 61 PRINT AT 1,0;"▌": PRINT AT
2,0;"▌": PRINT AT 3,0;"▌"
 62 PRINT AT 1,31;"▐": PRINT AT
 2,31;"▐": PRINT AT 3,31;"▐"
 65 PRINT AT 4,0;"
"
 66 PRINT AT 5,0;"▌": PRINT AT
6,0;"▌": PRINT AT 7,0;"▌": PRINT
 AT 8,0;"▌": PRINT AT 9,0;"▌": P
RINT AT 10,0;"▌": PRINT AT 11,0;
"▌": PRINT AT 12,0;"▌"
 70 PRINT AT 7,5;"Examen parcia
l de": PRINT AT 8,5;"Derecho Pro
cesal": PRINT AT 9,5;"Civil y Co
mercial."
 75 PRINT AT 14,0;"  Catedra  D
r.Luis M. Gaibrois  "
 76 PRINT AT 17,5;"Fecha: ";t$
 77 PRINT AT 20,0;"
"
 78 PRINT AT 21,0; FLASH 1;">":
 PRINT AT 21,1;"(Pulse 'ENTER').
"
 80 PAUSE 0: FLASH 0
 85 CLS
100 REM Pantalla Nombre y Apell
ido.
105 PRINT "
"
110 PRINT AT 3,2;"Por favor:":
PRINT AT 7,5;"TIPEE en el teclad
o de la": PRINT AT 9,3;"maquina
su NOMBRE Y APELLIDO.": PRINT AT
 19,5;"(luego, pulse 'ENTER')."
112 PRINT AT 21,0;"
"
114 PRINT AT 17,0;"
"
120 INPUT N$(A)
180 CLS
190 DEF FN O$(Z$)=Z$(I)+CHR$ 8+
CHR$ 21+CHR$ 1+"_"
192 LET Z$="INSTRUCCIONES:"
194 FOR I=1 TO LEN Z$
196 PRINT FN O$(Z$);
198 BEEP .10,20
199 NEXT I
205 REM Pantalla Instrucciones
.
207 PRINT AT 2,0;"■A continuaci
on apareceran en    pantalla las
 preguntas sobre    las que vers
a este examen.      ■Tambien figu
ran las respuestas."
209 PRINT AT 8,0;"■Tipee correc
tamente el numero   de la opcion
 que elige como     respuesta, p
orque no tiene      posibilidad
de corregir sus     errores."
220 PRINT AT 14,0;"■El teclado
de este ordenador    es el mismo
 de una maquina de   escribir.
.                    ■Comprendido
, comience!."
299 PRINT AT 20,0;"
"
300 PRINT AT 21,0;"(Para comenz
ar: pulse 'ENTER')."
310 PAUSE 0
320 FLASH 0: CLS
420 LET RA=0
440 LET RD=0
460 LET PN=0
990 PRINT AT 1,0;"
"
994 PRINT AT 20,0;"
"
995 PRINT AT 21,0;"(Pulse numer
o de opcion y ENTER)"
1085 GO SUB 8300
1500 GO SUB 8000
1502 PRINT AT 5,2;"(Art.330 CP)"
1505 PRINT AT 6,2;"El escrito de
 demanda debe        precisar el m
onto reclamado ?:"
1507 PRINT AT 14,5;"1) Si.": PRI
NT AT 16,5;"2) No."
1510 INPUT U$
1515 IF U$="0" THEN GO TO 8450
1520 IF U$="1" THEN GO SUB 8050
1525 IF U$="2" THEN GO SUB 8200
1530 IF U$<>"1" AND U$<>"2" THEN
 GO SUB 7900
1540 GO SUB 8300
1550 GO SUB 8000
1555 PRINT AT 5,2;"Hasta cuando
puede el actor    modificar la d
emanda ?:"
1557 PRINT AT 14,1;"1) hasta ant
es de abrirre el        juicio a
prueba.": PRINT AT 17,1;"2) hast
a antes de que la deman-    da s
ea notificada."
1560 INPUT U$
1565 IF U$="0" THEN GO TO 8450
1570 IF U$="2" THEN GO SUB 8050
1575 IF U$="1" THEN GO SUB 8200
1580 IF U$<>"2" AND U$<>"1" THEN
 GO SUB 7900
1590 GO SUB 8300
1600 GO SUB 8000
1610 PRINT AT 5,1;"El actor pued
e ampliar la cuan- tia de lo rec
lamado en la de-   manda, si ant
es de la senten-   cia vencen nu
evos periodos de   la obligacion
 ?:"
1613 PRINT AT 14,5;"1) Si.": PRI
NT AT 16,5;"2) No."
1615 INPUT U$
1620 IF U$="0" THEN GO TO 8450
1625 IF U$="1" THEN GO SUB 8050
1630 IF U$="2" THEN GO SUB 8200
1635 IF U$<>"1" AND U$<>"2" THEN
 GO SUB 7900
1640 GO SUB 8300
1650 GO SUB 8000
1655 PRINT AT 5,1;"Por cuantos d
ias se confiere     el traslado d
e la demanda en     el juicio ord
inario ?:"
1657 PRINT AT 14,5;"1) por 10 di
as.": PRINT AT 16,5;"2) por 15 d
ias.": PRINT AT 18,5;"3) por 20
dias."
1660 INPUT U$
1665 IF U$="0" THEN GO TO 8450
1670 IF U$="2" THEN GO SUB 8050
1675 IF U$="1" OR U$="3" THEN GO
 SUB 8200
1680 IF U$<>"1" AND U$<>"2" AND
U$<>"3" THEN GO SUB 7900
1685 GO SUB 8300
1700 GO SUB 8000
4305 PRINT AT 5,2;"(Art.348 CP)"
4308 PRINT AT 6,2;"Si el actor n
o tiene domicilio  pero tiene in
muebles en la Re-  publica, corr
esponde la excep-  cion de arrai
go ?:"
4310 PRINT AT 14,5;"1) Si.": PRI
NT AT 16,5;"2) No."
4315 INPUT U$
4320 IF U$="0" THEN GO TO 8450
4325 IF U$="2" THEN GO SUB 8050
4330 IF U$="1" THEN GO SUB 8200
4335 IF U$<>"1" AND U$<>"2" THEN
 GO SUB 7900
7721 GO TO 8500
7900 PRINT AT 12,0; FLASH 1;" TI
PEO MAL:PRESTE MAS ATENCION!."
7915 PAUSE 150: FLASH 0
7930 RETURN
8000 REM = Subrutina contador de
 preguntas.
8010 LET PN=PN+1
8015 PRINT AT 2,10;"Pregunta: ";
PN
8020 RETURN
8050 REM = Subrutina contador de
preguntas correctas.
8055 LET RA=RA+1
8060 PRINT AT 0,0; FLASH 1;"   C
ORRECTO   "
8065 PAUSE 170
8070 PRINT AT 0,1; FLASH 0;"
"
8075 PRINT AT 0,0;"Correctas: ";
RA
8085 PAUSE 80
8090 RETURN
8200 REM = Subrutina contador de
 respuestas incorrectas.
8205 LET RD=RD+1
8210 PRINT AT 0,17; FLASH 1;"  I
NCORRECTO  "
8215 PAUSE 170
8220 PRINT AT 0,17; FLASH 0;"
"
8225 PRINT AT 0,18;"Incorrectas:
";RD
8235 RETURN
8300 REM = Subrutina borrado de
antalla,(de linea 5 a linea 21).
8304 PRINT AT 4,0;"
"
8305 PRINT AT 5,0;"
"
8310 PRINT AT 10,0;"
"
8315 PRINT AT 15,0;"
"
8330 RETURN
8450 CLS
8455 PRINT AT 10,10;"SUFICIENTE"
8456 PRINT AT 19,0;"
"
8457 PRINT AT 20,0;" (Para segui
r: pulse 'ENTER')."
8460 PAUSE 0
8500 REM = Pantalla final.
8505 CLS
8510 PRINT "
"
8515 PRINT AT 1,1;"FACULTAD DE C
IENCIAS JURIDICAS"
8520 PRINT AT 4,0;"
"
8525 PRINT AT 6,1;"El alumno ";N
$(A)
8529 PRINT AT 7,1;"ha rendido en
 la fecha examen   parcial de la
 asignatura Dere-  cho Procesal
Civil y Com., ha-"
8530 PRINT AT 10,1;"biendo respo
ndido  ";RA;"  veces en  forma c
orrecta, y  ";RD;"  veces en  fo
rma incorrecta."
8532 PRINT AT 15,0;t$
8535 PRINT AT 17,13;"
"
8540 PRINT AT 18,13;"Dr.Luis M.G
aibrois"
8545 PRINT AT 20,0;"(Sr.Profesor
: ingrese la cali-  ficacion del
 alumno y ENTER)."
8550 INPUT C(A)
8551 CLS
8552 PRINT AT 5,2;"El alumno ";
N$(A);"
        ha obtenido la calificac
ion
        de  ";C(A)
8553 PRINT AT 21,1;"(Para seguir
: pulse 'ENTER')."
8558 PAUSE 0
8560 IF A<X THEN GO TO 35: CLS
8565 CLS
8570 FOR V=1 TO X
8580 PRINT N$(V);" ";C(V)
8590 NEXT V
```

PANTALLA

```
■A continuacion apareceran en
pantalla las preguntas sobre
las que versa este examen.

■Tambien figuran las respuestas.

■Tipee correctamente el numero
de la opcion que elige como
respuesta, porque no tiene
posibilidad de corregir sus
errores.

■El teclado de este ordenador
es el mismo de una maquina de
escribir.

■Comprendido, comience!.
```






19 Programas Inéditos

Desarrollos:

Convertimos la TS 2068

Interface de Grabador
Para Commodore C64

Segundo Concurso:
Importantes Premios